

# Juez Automático para la Evaluación de Problemas de Programación en los Primeros Cursos de Estudios de Informática

---



Trabajo de Fin de Grado, Facultad de Informática, Universidad Complutense  
de Madrid

Curso 2012-2013

Realizado por  
Jéssica Martín Jabón, Javier Martín Moreno-Manzanaro y Pablo Suárez Díaz

Proyecto dirigido por  
Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín

# Autorización

Los abajo firmantes Jéssica Martín Jabón, Javier Martín Moreno-Manzanaro y Pablo Suárez Díaz, matriculados en el TRABAJO DE FIN DE GRADO de la Facultad de Informática, autorizamos a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la presente memoria: “Juez Evaluador de Problemas de Programación”, como el código, los contenidos audiovisuales, la documentación y el prototipo desarrollado durante el curso académico 2012-2013 bajo la dirección de los profesores Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Jéssica Martín Jabón  
46928476-N

Javier Martín Moreno-Manzanaro  
50231006-H

Pablo Suárez Díaz  
X3612644B

En Madrid a 21 de junio de 2013

# Agradecimientos

Para nosotros, la realización de este proyecto ha supuesto un reto así como el inicio de un camino de autoaprendizaje e investigación. Sin embargo, la satisfacción de haber alcanzado nuestros objetivos (o al menos algunos de ellos), compensa el esfuerzo empleado. La consecución de esa satisfacción, y la madurez adquirida en este y otros proyectos, se la debemos en parte a los profesores (*casi* todos) que hemos tenido a lo largo de nuestro periplo universitario, y muy especialmente a nuestros directores de este Trabajo de Fin de Grado, Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín, cuya guía ha sido imprescindible.

También tenemos que agradecer a varios compañeros, sus opiniones y críticas (constructivas en la mayoría de casos), que nos echaron una mano para identificar algunos fallos y corregirlos en pos de una mejora de nuestro proyecto.

Por último, y aunque sirva para terminar el tópico que forman estas líneas, queremos dar las gracias a nuestras familias. A nuestros padres, hermanos y otros allegados, que han ocupado los roles de mecenas, cuidadores, animadores y muchos otros con tal de brindarnos apoyo en nuestro camino hasta el final de manera incondicional.

Muchas gracias a todos.

Pablo, Jéssica y Javier.

# Resumen

Muchos de los participantes en concursos de programación, entrenan resolviendo problemas típicos de libros de texto, sin embargo, cada vez más, se impone la necesidad de acceso a problemas y evaluaciones de los mismos, a través de la red de redes. En este proyecto, tratamos de resolver esa necesidad, así como de conseguir un mayor calado académico al usarlo como herramienta en un entorno docente real, con usuarios reales, que puedan realizar esos entrenamientos a los que nos referíamos de manera fácil y cómoda, interactuando simplemente con un interfaz web.

## Palabras Clave

AJAX  
cURL  
Desafíos de Programación  
Juez Automático  
Juez Evaluador  
Juez On-line  
JSON  
PHP  
Servicios web  
Web

# Abstract

Many contestants in programming competitions are trained by solving typical algorithm problems from their books, due to this, it is needed a platform through the world wide web that allows users to reach for these exercises and check their skills on them by keeping statistics of the currently solved problems. This project is about to change a reality, to accomplish more student implication by using it as a significant tool. One of our goals is to provide to Universidad Complutense de Madrid with an implement in a learning environment, rising this institution to the same level of other important Universities that have their own online-judges.

## Keywords

AJAX  
cURL  
Programming challenges  
Automatic Judge  
Evaluator Judge  
On-line Judge  
JSON  
PHP  
Web Services  
Web

*- "No hay preguntas sin respuestas, sólo preguntas mal formuladas" -*  
**Morfeo en TheMatrix**

# ÍNDICE

## Contenido

|   |    |
|---|----|
| 1.- Introducción .....  | 9  |
| 1.1.- Justificación .....   | 9  |
| 1.2.- Objetivos .....   | 9  |
| 1.2.1. - Expectativas de uso docente en la UCM .....  | 10 |
| 1.2.2. - Expectativas de uso externo .....  | 10 |
| 1.3. - Resumen.....   | 11 |
| 2-Estado del arte .....   | 12 |
| 2.1. - UVa OnlineJudge [uHunt] ( <a href="http://uva.onlinejudge.org">http://uva.onlinejudge.org</a> ) .....                      | 12 |
| 2.1.1. - Necesidad de registrar .....   | 12 |
| 2.1.2. - Distribución interfaz .....  | 13 |
| 2.1.3. - Ver envíos en tiempo real .....  | 14 |
| 2.1.4. - Clasificación de los problemas .....   | 15 |
| 2.2. - Light OJ ( <a href="http://www.lightoj.com">www.lightoj.com</a> ) .....  | 16 |
| 2.2.1. - Necesidad de registrar .....   | 16 |
| 2.2.2. - Distribución interfaz .....  | 16 |
| 2.2.3. - Ver envíos en tiempo real .....  | 18 |
| 2.2.4. - Clasificación de los problemas .....   | 18 |
| 2.3. - Sphere online judge ( <a href="http://www.spoj.pl">http://www.spoj.pl</a> ) .....  | 19 |
| 2.3.1. - Necesidad de registrar .....   | 19 |
| 2.3.2. - Distribución interfaz .....  | 20 |
| 2.3.3. - Ver envíos en tiempo real .....  | 21 |
| 2.3.4. - Clasificación de los problemas .....   | 21 |
| 2.4. - Olimpiada Informática Española ( <a href="http://olimpiada-informatica.org/">http://olimpiada-informatica.org/</a> ) ..... | 22 |
| 2.4.1. - Necesidad de registrar .....   | 22 |
| 2.4.2. - Distribución interfaz .....  | 23 |
| 2.4.3. - Ver envíos en tiempo real .....  | 23 |
| 2.4.4. - Clasificación de los problemas .....   | 23 |
| 2.5 Jutge.org ( <a href="http://www.jutge.org">www.jutge.org</a> ) .....  | 25 |
| 2.5.1 Distribución de la página .....   | 25 |
| 2.5.2 Envíos en tiempo real.....  | 26 |
| 2.6 FLOP (Free Laboratory Of Programming).....  | 27 |

|  |    |
|--|----|
| 3. - Arquitectura del Sistema.....   | 29 |
| 3.1. Servicios Web .....   | 29 |
| 3.2. PHP + JavaScript .....  | 29 |
| 3.3 TOMCAT .....   | 31 |
| 4.- Estudio de Navegación.....   | 32 |
| 4.1. - Distribución general.....   | 32 |
| 4.2. - Estadísticas (Últimos Envíos).....  | 34 |
| 4.2.1. - Código enviado.....   | 35 |
| 4.3. - Problemas .....   | 36 |
| 4.3.1. - Lista de Volúmenes .....  | 36 |
| 4.3.2. - Lista de Problemas .....  | 37 |
| 4.3.3. - “Mini-site” de un problema .....  | 37 |
| 5. - Implementación .....  | 41 |
| 5.1. - jQuery.....   | 41 |
| 5.1.2. Características (solo destacaremos las utilizadas en este proyecto) ..... | 42 |
| 5.1.3. Uso.....  | 42 |
| 5.1.4. Función \$().....   | 42 |
| 5.1.5. Función \$.getJSON (). .....  | 44 |
| 5.2. AJAX.....   | 45 |
| 5.2.1. Tecnologías incluidas en Ajax.....  | 46 |
| 5.3. GeSHi – Generic Syntax Highlighter (Marcador de sintaxis genérico) .....    | 47 |
| 5.4. CodeMirror .....  | 49 |
| 5.5. VanadiumJS .....  | 50 |
| 5.6. Diagramas de google .....   | 52 |
| 5.7. Cross-site scripting .....  | 54 |
| 5.8. cURL.....   | 57 |
| 5.9. MathJax .....   | 58 |
| 5.10. RECAPTCHA .....  | 59 |
| 6. – Conclusiones.....   | 62 |
| 6.1. – Opiniones y valoraciones .....  | 62 |
| 6.2. – Valoración personal.....  | 66 |
| 7. – Bibliografía.....   | 68 |



# 1.- Introducción

En primer lugar nos gustaría explicar los motivos por los cuales nos sentimos atraídos por este proyecto. La razón principal fue la necesidad de cursar los créditos obligatorios del Trabajo de Fin de Grado (en futuras alusiones TFG); de entre los proyectos que se ofertaban, lo primero que nos llamó la atención fueron los directores de proyecto Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín. Nos entusiasmó la idea de trabajar con tan cualificados profesores con los que ya habíamos tenido trato en el pasado. En segundo lugar cuando compartieron con nosotros su idea del proyecto, la idea nos motivó aún más, ya que no sólo tendríamos la oportunidad de trabajar mano a mano con ellos sino que además podríamos contribuir con la facultad. Por último, la posibilidad de adentrarnos en las tecnologías web, nos llamó especialmente la atención, dado el uso y la proyección de futuro que tienen estas tecnologías.

## 1.1.- Justificación

Nuestro proyecto pretende ser una fuente de conocimiento para aquellas personas que se inician en el mundo de la algoritmia. Pese a que este tipo de sistema no es novedoso en su funcionamiento, sí lo es al tipo de persona al que va dirigido. Normalmente los jueces de problemas, como el que hemos desarrollado, están destinados al entrenamiento para concursos; sin embargo nuestra aplicación sirve para aprender, con unos problemas cuyo nivel es más asequible y adecuado. De este modo también se pretende conseguir un efecto positivo que mantendrá a la gente motivada, teniendo en cuenta que serán personas que se están iniciando en este tipo de conocimiento.

## 1.2.- Objetivos

El objetivo fundamental de este proyecto es conseguir un juez de resolución de problemas que se pueda aplicar al ámbito docente, y no sólo a concursos de programación; es decir conseguir una herramienta de apoyo en el proceso de aprendizaje necesario para formarse en el campo de la resolución de problemas de programación. Así pues, a la evaluación de problemas, se añaden ciertas funciones como la descripción de los errores o la aportación de pistas que ayuden a alcanzar la solución correcta. Además, debido precisamente al carácter docente con el que se impregna la aplicación es especialmente útil, para ser usada en asignaturas de programación en el entorno universitario, como de hecho se pretende hacer con algunos cursos de la Facultad de Informática en la asignatura de EDA (Estructuras de Datos y Algoritmos).

### 1.2.1. - Expectativas de uso docente en la UCM

El sistema está dirigido a estudiantes de asignaturas de programación. Son muchos los títulos de grado en los que existen este tipo de asignaturas. Por mencionar únicamente las de la propia facultad de Informática en donde se ha desarrollado este TFG, existen cuatro asignaturas en donde claramente se podría utilizar el sistema. Por hacernos una idea del número de usuarios/alumnos potenciales, la tabla siguiente muestra el número de grupos y alumnos aproximados que tiene cada una de ellas en el curso 2011/2012.

| Asignaturas  | Grupos | Alumnos (aprox.) |
|--|--------|------------------|
| Fundamentos de la programación                     | 8      | 600              |
| Estructuras de Datos y Algoritmos                  | 5      | 300              |
| Tecnología de la Programación                      | 5      | 300              |
| Métodos Algorítmicos en la Resolución de Problemas | 4      | 80               |

Las previsiones, en cuanto al número de alumnos son optimistas y se espera que aumenten en cursos sucesivos.

### 1.2.2. - Expectativas de uso externo

Además de los alumnos de asignaturas de programación, existen otros posibles usuarios de nuestro sistema: son los usuarios que se preparan para concursos de programación y que no tienen por qué pertenecer a ninguna institución universitaria.

Desde un principio uno de los motores que impulsó la realización de este proyecto fue su utilización para el entrenamiento de los participantes en el concurso de programación ProgramaMe.



ProgramaMe es el concurso de programación para alumnos matriculados en Ciclos Formativos de Formación Profesional. No se establecen diferencias entre los diferentes ciclos existentes, por lo que se permite la participación de cualquier alumno que curse Formación Profesional en el territorio español.

Los alumnos participan en equipos formados por tres estudiantes. Durante el concurso, los participantes disponen de 4 horas para resolver entre 8 y 12 problemas, siendo 10 el número más habitual. Deben programar sus soluciones en C, C++ o Java.

Los problemas son comprobados por un *juez automático* que los ejecuta de manera autónoma contra un conjunto de casos de prueba establecidos de antemano. El juez, supervisado por

jueces humanos, proporciona un veredicto sobre el problema, aceptándolo o marcándolo como incorrecto, pero en ningún caso ofrece información sobre cuál es el error. En caso de que el veredicto sea incorrecto el equipo puede realizar envíos posteriores del mismo problema para intentar corregirlo.

Actualmente los problemas para prepararse para este reto están disponibles en la página web en formato pdf.

**Problemas del concurso**

El concurso constará de entre ocho y diez problemas de los más diversos temas. A continuación os dejamos los utilizados durante los concursos realizados hasta la fecha en formato electrónico para que podáis practicar con ellos.

**Concursos de 2011**

[Cuadernillo de problemas ProgramaMe 2011](#)

**Concursos de 2012**

[Cuadernillo de problemas del Regional de Madrid de Marzo de 2012](#)

[Cuadernillo de problemas del Nacional ProgramaMe 2012](#)

**Concursos de 2013**

[Cuadernillo de problemas del On Line de ProgramaMe 2013 en la edición de mañana](#)

[Cuadernillo de problemas del On Line de ProgramaMe 2013 en la edición de tarde](#)

[Cuadernillo de problemas de los Regionales de Madrid y Reus de ProgramaMe 2013](#)

[Cuadernillo de problemas del Regional de Zaragoza de ProgramaMe 2013](#)

### 1.3. - Resumen

Así pues, lo que podrás encontrar en la presente memoria, podría dividirse claramente por relevancia según el perfil del lector.

**Para Usuarios:** Exploramos la navegabilidad de la aplicación en el punto de *Estudio de Navegación*, en el que ahondamos en las distintas páginas viendo cómo se relacionan entre sí. También puede ser interesante para familiarizarse con el dominio de la aplicación *El Estado del Arte*, en el que destacamos las características de otros jueces evaluadores.

**Para programadores:** Dejamos constancia de aspectos técnicos de codificación y estructuración en los apartados reservados para *Implementación y Arquitectura del Sistema*.

## 2-Estado del arte

Las webs que contienen jueces evaluadores de problemas no son tan escasas como podríamos llegar a pensar, por ello es importante ver algunos de ellos centrándonos en puntos importantes.

### 2.1. - UVaOnlineJudge [uHunt] (<http://uva.onlinejudge.org>)

Este es el juez de la universidad de Valladolid y principal inspiración para nuestro proyecto. Su sencillez hace que sea bastante intuitivo de usar y por eso hemos decidido adaptar ciertos detalles a nuestra interfaz.

#### 2.1.1. - Necesidad de registrar

No es necesario estar registrado para ver el contenido de la página, puedes ver los problemas o los últimos envíos, entre otras cosas.

Sólo es necesario estar registrado para tener acceso a las opciones de:

- **Enviar un problema**, para ello sólo será necesario seleccionar el tipo de lenguaje, el id del problema y el código del mismo, que podrá escribirse en un cuadro de texto o subirse directamente desde un fichero de tu ordenador.
- **Ver/modificar tu perfil**, en este apartado se muestran los datos que proporcionamos al crear la cuenta, y se da la opción de cambiarlos. También se da la posibilidad de subir una foto para mostrarla en el perfil.
- **Ver tus estadísticas**. Proporciona datos del usuario tales como estadísticas con gráficos en forma de tarta, número de envíos totales, número problemas intentados y resueltos, fecha del primer y del último envío y una tabla con todos los problemas que has resuelto.
- **Ver tus últimos envíos**, en esta página se muestran los últimos envíos que el usuario ha realizado, proporcionando además del ID del envío, el ID del problema, el título, el veredicto, el lenguaje, el tiempo de ejecución y la fecha de envío.

##### 2.1.1.1. - Formulario de registro

Para registrarse en la página los datos que se solicitan son los típicos: nombre, email, nick y contraseña, estos dos últimos campos son los necesarios para loguearse. Además puedes marcar la opción de que te manden los veredictos por email y carecen de seguridad para evitar que un bot cree cuentas, como captchas o cualquier otro tipo de mecanismo que lo evite. También se informa de qué campos se mostrarán en el perfil.

### 2.1.2. - Distribución interfaz

La portada de la web se divide en tres columnas. Esta distribución se modificará al navegar por la web dejando la columna de la izquierda fija y usando el resto del espacio según el contenido que se muestre.

Precisamente, esta columna izquierda es especialmente destacable, por mantener secciones especialmente relevantes, tales como:

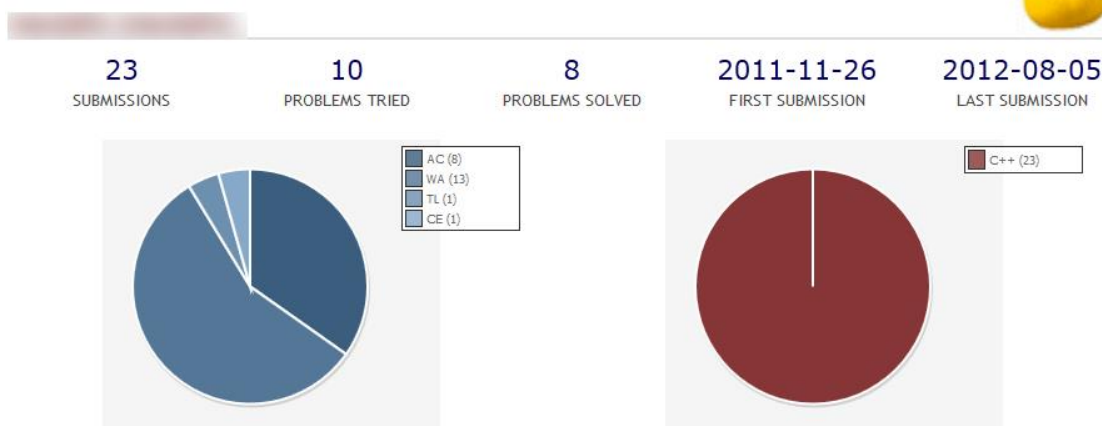
- Login: En esta zona se mantiene un pequeño formulario para loguearse (o registrarse), solicitando Nick y contraseña.
- Un cuadro de búsqueda de Google
- Menú principal: En esta sección se muestra una breve colección de enlaces a la portada (Home), al formulario de contacto, a la antigua web de la UVA y a la página con los problemas de las finales regionales y mundiales
- Juez online: Muestra una colección de enlaces referentes al juez. El primero, a uHunt, y el resto a páginas con información de estadísticas (últimos envíos, lenguajes usados, problemas intentados por los usuarios, etc.), concursos, problemas y envíos, de los cuáles se dan los datos típicos como el problema, el resultado de la evaluación, consumo de memoria, fecha, etc. Además de todo esto, podemos encontrar, un foro, una página de ayuda, y enlaces a redes sociales, que sirven de ayuda a los usuarios novatos.
- Enlaces a webs colaboradoras. La parte central muestra un mensaje de bienvenida y anuncios relacionados con la web, como libros de resolución de problemas. Esta parte central es la empleada en mostrar la información al navegar por otras páginas absorbiendo el espacio de la columna derecha.

El lateral derecho contiene la portada de un libro (a modo de anuncio) y la lista de los próximos concursos.

Cabe destacar que la distribución de los menús y submenús así como la ubicación de los mismos hacen que la navegación por la página sea sencilla e intuitiva.

Salta a la vista, no obstante, que el sistema tiene ya varios años y que no goza del dinamismo al que hoy día estamos acostumbrados con las aplicaciones Web: todo el contenido es generado en el servidor y el navegador lo único que hace es, en contadas ocasiones, refrescar la página para actualizar la información que contiene. Eso hace que sea incómodo ver, por ejemplo, el estado en tiempo real del servidor mostrando los envíos pendientes de evaluar, etc. Otro ejemplo es el de los propios envíos: tras subir la solución de un problema, se puede ir a la página de "mis envíos" para conocer el veredicto. Sin embargo, una vez que la página se carga, si el juez aún no la ha evaluado (y por tanto su estado es "In queue"), habrá que actualizar la página una y otra vez hasta conseguir el veredicto.

## User Statistics



## Solved problems

| Problem | Ranking | Submission | Date                | Run time |
|---------|---------|------------|---------------------|----------|
| 100     | 8042    | 10437715   | 2012-08-05 15:09:33 | 0.380    |
| 514     | 797     | 9829217    | 2012-03-07 22:37:02 | 0.228    |
| 10004   | 5146    | 9969295    | 2012-04-08 11:03:28 | 0.008    |
| 10055   | 14898   | 9509452    | 2011-11-26 19:31:23 | 0.492    |
| 10107   | 837     | 9884238    | 2012-03-19 22:06:12 | 0.040    |
| 10469   | 5541    | 9543617    | 2011-12-08 14:58:35 | 0.012    |
| 11496   | 318     | 9585109    | 2011-12-22 23:46:53 | 0.024    |
| 12015   | 634     | 9865023    | 2012-03-15 22:52:46 | 0.008    |

## 2.1.3. - Ver envíos en tiempo real


Para luchar contra el poco dinamismo del portal de la UVa, surgió un desarrollo paralelo, uHunt, que hace crawling continuo de la página de la UVa para presentar al usuario una página dinámica que se actualiza periódicamente mostrando los envíos del propio usuario y el estado del servidor<sup>1</sup>. Es decir, permite ver los envíos en tiempo real a través de una tabla, con los campos: id del envío, id y título del problema, enlace al tema del foro para hablar sobre el problema, nick del usuario, veredicto (para hacerlo más intuitivo se apoyan en colores, verde para aceptado, amarillo para error de compilación, etc.), lenguaje de programación utilizado, tiempo de ejecución, el mejor tiempo de ese problema hasta el momento, el ranking y cuánto tiempo ha pasado desde el envío.

Además puedes elegir el número de envíos que quieres ver en la tabla ofreciendo listas de 5, 10, 20, 50 ó 100 envíos.

El sistema, además, incorpora un ranking de dificultad de los problemas en base al número de usuarios distintos que han sido capaces de resolverlos, y ayuda a seleccionar el siguiente ejercicio que debes hacer, en base a los ejercicios que tienes resueltos. La guía no utiliza

<sup>1</sup> Posteriormente, los administradores del sistema de la UVa dieron permiso de lectura a algunas tablas de su base de datos al desarrollador de uHunt, de forma que actualmente éste funciona sin necesidad de hacer *crawling*.

ningún mecanismo sofisticado (basado en categorías de problemas o algo similar) sino simples estadísticas de problemas aceptados, etc.



## uVa uHunt

hunt problems that matter

uHunt is a tool for [Uva online-judge](#) that keeps statistics, provide selections of problems to solve, and exposes a web API for other web developers to build upon it.

To quickly submit your solution, use [Uva Quick Submit](#).

Your Uva username :

Search Problem Number :

```
-- Users(9/113):1d
felix_halim:1d
R00Dv:18h
Safait:4h
tbnjha:53m
ahmed.m3:0m
gaarav999:1d
bonsu:2h
Mr.Bean:6h
veerweapon:1h
```

[\[ Sign In \]](#)

post your message here (max 255 chars)

| Live Submissions (hide) |       |                       |   |              |        |       |       |      |             | Show : 5   10   25   50   100 |  |  |  |  |
|-------------------------|-------|-----------------------|---|--------------|--------|-------|-------|------|-------------|-------------------------------|--|--|--|--|
| #                       |       | Problem Title         | User (username)   | Verdict      | Lang   | Time  | Best  | Rank | Submit Time |                               |  |  |  |  |
| 11922995                | 12348 | Fun Coloring          | <a href="#">discuss</a> Karolis Kusas (shadows4)          | - In queue - | C++    | 0.000 | 0.008 | -    | 4 secs ago  |                               |  |  |  |  |
| 11922994                | 12085 | Mobile Casanova       | <a href="#">discuss</a> Vu Van Tuyen - HUBT University... | - In queue - | C++    | 0.000 | 0.012 | -    | 36 secs ago |                               |  |  |  |  |
| 11922993                | 583   | Prime Factors         | <a href="#">discuss</a> Fahmi (fahmitok)                  | - In queue - | C++    | 0.000 | 0.048 | -    | 42 secs ago |                               |  |  |  |  |
| 11922992                | 11418 | Clever Naming Patt... | <a href="#">discuss</a> PCCA_ (PCCA_)                     | Wrong answer | C++    | 0.026 | 0.000 | -    | 50 secs ago |                               |  |  |  |  |
| 11922991                | 10055 | Hashmat the Brave...  | <a href="#">discuss</a> Feng Xiong (messiforever)         | - In queue - | ANSI C | 0.000 | 0.000 | -    | 1 mins ago  |                               |  |  |  |  |

### What is uHunt

[uHunt](#) is my personal (self-financed) project. It is a tool that helps me (and others) to find interesting algorithm problems to solve at [Uva Online Judge](#), as well as keeping statistics of the currently solved problems. uHunt also exposes web API for other web developers that are interested in querying Uva submissions data and build their own tools.

### How uHunt was invented

I love solving programming contest (algorithm) problems. Back at year 2000, the only authoritative Online Judge that hosts good quality of past programming contest problems was the [Uva Online Judge](#). However, as a newbie at that time, I was confused and frustrated in picking the next problem to solve. I want the next problem to match my current skill (i.e, not too hard). That was what triggered me to build a tool (a website) to organize the problems according to the difficulty level. At that time I only use the tool for myself, but over time, people around the world began to recognize the website and started using it.

### How uHunt evolved over time (development challenges)

I keep using the tool myself and improve it bit by bit whenever I have spare time. Along the way, I learned a few web programming technologies like HTML, Javascript, PHP, MySQL. However, as the number of **concurrent users** accessing the website grows to more than 100 per day, the (shared) server began to crumble. This is because the tool requires high interactivity. It tries to display the most up-to-date statistics and provide chat functionality for small discussions. The Apache web server was not good enough to handle such workload. Fortunately, a new platform emerged ([node.js](#)) that solved the [c10k problem](#) which allows a regular server to handle thousands of concurrent clients with far smaller memory footprint. This was the perfect platform to switch to. Building uHunt as a node.js application was easy as it uses Javascript which I'm familiar about. However, there was another problem, the database (MySQL) was also beginning to crumble as the number of rows in the submission table went close to 10 millions (Uva OJ has that many submissions and growing). The time to update/maintain the indexes is so high that it increases the latency (responsiveness) of the website. I began to search for new database technology and stumbled upon a new exciting adaptive indexing technology called [Database Cracking](#). It was perfect for my needs as it can process a large amount of tuples with low latency for each individual query. Moreover, it quickly gets faster (adapt) as more queries are performed! I built my own version of the Database Cracking using C++ and connect it to my node.js application. Since then, uHunt has been running fast and attracting even more users (reaching 200+ concurrent users per day). Thanks to the Uva admin for granting uHunt a direct database access which allows uHunt to have a timely response to user submissions. With these powerful technologies, it does scale well and allows me to expose uHunt as a web API for other tools to build upon. The web API is like a JSON data feed that can be used by other web developers who are interested in building variety of other tools for Uva OJ such as virtual contests, custom ranklists, or any other statistical tools.

### Who has been using uHunt

uHunt has been heavily used by the [National University of Singapore](#) to train computer science students for ACM ICPC. The popularity of uHunt increases even more when universities around the world included [uHunt](#) and the [Competitive Programming](#) book in their [list of references](#). The book exercises are integrated with uHunt for self-training.

It has been an amazing experience to see uHunt grows :)

## 2.1.4. - Clasificación de los problemas

Los problemas se clasifican por categorías, según sean problemas normales o de concurso. Estos últimos se agrupan además por el concurso en el que se plantearon. Cada una de estas categorías puede estar organizada por volúmenes (compuestos por 100 problemas), por autor, por año o por título.

Como comentario personal, la navegación es simple, no obstante la agrupación de los volúmenes y su distribución no es tan intuitiva cuando quieres buscarlos por otro parámetro que no sea el id.

## 2.2. - Light OJ ([www.lightoj.com](http://www.lightoj.com))

Este juez evaluador de problemas fue creado por Jane AlamJan. El lenguaje que se usa en esta página es el inglés, y no dispone de ninguna opción para cambiarlo.

### 2.2.1. - Necesidad de registrar

Para tener acceso al contenido de la paginas, es necesario estar registrado y logueado, ya que de otro modo no se puede ver ningún tipo de información, ni si quiera se da la opción de ver la lista de problemas. Tan sólo aparecen dos cuadros de texto (Nick/mail y contraseña) y un enlace para ir a un formulario de registro.

Como usuario es poco amigable el hecho de que no te deje navegar por la página, puesto que si estás buscando un juez evaluador de problemas, antes de ceder tus datos es preferible ver el tipo de problemas que tiene, servicios que pueden usarse, etc.

Light OJ

Server Time: Wed Aug 1, 2012 4:36 am

### Member Login

User Id or E-mail

Password

☐ Remember Me

Login

New user? [Register here](#). Or contact us at [lightoj.mod\[AT\]gmail\[DOT\]com](mailto:lightoj.mod[AT]gmail[DOT]com)

Some Bangladeshi Contests that are announced in UVA site will take place in this site.  
But users need to have a new account in LightOJ. (Courtesy of UVA)

Use [Google Chrome](#) for Better View

Developed and Maintained by  
JANE ALAM JAN

Copyright © 2012  
LightOJ, Jane Alam Jan

#### 2.2.1.1. – Formulario de registro

Para registrarnos en la página web es necesario introducir el nombre (no más de 100 caracteres), el nombre de una universidad (no más de 100 caracteres), seleccionar un país (están en inglés ordenados alfabéticamente), el email y para comprobar que no eres un bot te hacen una pregunta que es una operación numérica del estilo “¿Cuál debería ser el resultado de sumar 72 y 68?” para responder a la pregunta hay un campo de texto en el que se escribe la respuesta.

### 2.2.2. - Distribución interfaz

La página consta de dos menús uno en la parte superior y otro lateral cuyas opciones cambian según la opción que hayas seleccionado en el menú superior.



Opciones del menú superior:

Home: muestra la página de inicio, con un mensaje de bienvenida, también puedes ver tu perfil.

Volume: permite ver los problemas (organizados por volúmenes), categorías, enviar un problema o ver tus envíos

Contest: permite ver los concursos que están abiertos en ese momento además de los concursos para coger práctica.

Training: para entrenar.

Community: permite ver los foros.

Credits: muestra quién lo ha desarrollado y unos agradecimientos.

Para ver el recorrido por el menú lateral hablaré de las opciones que haya dependiendo de la opción del menú superior que se haya marcado.

#### - Home

El menú lateral se divide en dos submenús. De la primera mitad las opciones más relevantes son:

MyInfo: muestra un formulario en el que se permiten cambiar los datos que introdujiste en el registro, además de añadir más información como el lenguaje de programación, la zona horaria, la institución, el país, etc.

Logout: Un enlace que permite dejar de estar autenticado.

La segunda mitad contiene las opciones referentes al manejo de los mensajes que se pueden intercambiar los usuarios entre sí, y no vamos a profundizar en ellas.

#### - Volume

En este submenú las opciones más destacables son:

**Problem set:** muestra una tabla con los problemas, agrupados por volúmenes.

**Submit problem:** muestra el formulario para enviar un problema, en el que hay que rellenar un campo para identificar el problema y el lenguaje de programación, que se rellena automáticamente con la opción que seleccionaste en tu perfil. Para enviar el código sólo es posible a través de un cuadro de texto, destinado a tal efecto.

**My submission:** muestra una tabla con los últimos 20 envíos del usuario. Las columnas de dicha tabla son: id del envío, fecha del envío, título del problema, lenguaje de programación, uso de la CPU, uso de la memoria y veredicto.

**Judge status:** muestra los últimos 50 envíos de todo el juez.

**Ranklist:** da la opción de ver diferentes rankings ordenados por usuarios. Hay varios rankings, como el de usuarios que más problemas han resuelto (en la tabla se muestra la posición del ranking, el nombre del usuario, el número de problemas resueltos y el número de problemas intentados), el de número de problemas resueltos por países, el de número de problemas resueltos por tus amigos (es igual que la tabla de los usuarios con más problemas resueltos pero sólo con los usuarios que tengas marcados como amigos).

**Userstatistics:** muestra las estadísticas de tu usuario. En ellas aparecen los datos de tu perfil y dos tablas; una con el número de problemas resueltos, problemas enviados, total de envíos, el número de mensajes en el foro y la posición del ranking que ocupas; además de mostrar una gráfica de barras con los veredictos de tus problemas. La otra tabla muestra una lista con los id de los problemas resueltos.

### 2.2.3. - Ver envíos en tiempo real

Puedes ver los envíos que los usuarios están realizando en tiempo real, para ello se muestra una tabla con los campos de id del envío, fecha del envío, nombre del usuario, título del problema, lenguaje de programación, uso de la CPU, uso de la memoria y veredicto.

### 2.2.4. - Clasificación de los problemas

Los problemas están agrupados, por defecto, en volúmenes de 100 problemas cada uno pudiendo verlos además agrupados por categoría o por autor. Lo más destacable de esta última agrupación es que si seleccionamos un autor nos muestra una lista con todos los problemas que ha creado.

## 2.3. - Sphere online judge (<http://www.spoj.pl>)

Este juez es bastante amigable, el idioma que usa es el inglés y no dispone de ninguna opción para cambiarlo por otro. Es destacable la cantidad de publicidad que hay, en todas sus páginas, y además no está relacionada con el tema de los algoritmos

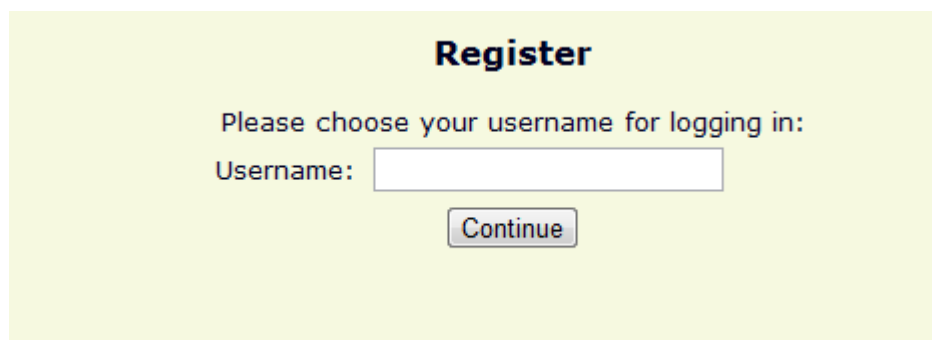
### 2.3.1. - Necesidad de registrar

Es necesario autenticarse si se quiere enviar un problema, pero para todo lo demás la web te deja que navegues por ella, de hecho incluso la opción de submit la puedes pulsar y ves el formulario de envío en el que hay un cuadro de texto para escribir el código, también se puede seleccionar un archivo que tengas en el ordenador, seleccionar el lenguaje y escribir el código del problema. Pero si no estás registrado y le das a enviar te redirige a una página en la que se te recomienda crearte una cuenta y en la columna de la izquierda aparece un cuadro de login.

Además también aparece en el menú la opción de ver tu perfil, la cual nos lleva a las estadísticas de nuestro usuario tales como la posición en el ranking que ocupa, los puntos conseguidos, el nick, el país, y una tabla con el número de problemas resueltos, enviados, aceptados, el número de problemas wrong answer, con error de compilación, con error en tiempo de ejecución y con tiempo límite excedido. En esta misma página también dispones de varios enlaces, uno para modificar tus datos de usuario, además también puedes pedirles trabajo y ver el historial de tus envíos.

#### 2.3.1.1. – Formulario de registro

El formulario de registro es diferente a como habitualmente son los formularios de registro. Al pulsar en la opción de registro no redirige a una página en la que sólo se nos pide un nombre de usuario. Una vez has escrito un usuario válido le das a “Continue” y te redirige a otra página en la que hay más campos que rellenar, menos el de elegir un nick que ya se eligió en la página anterior y te lo muestran, el resto de campos son: tu nombre, el email, la contraseña, el país, el lenguaje de programación, la institución y la url, aunque estos dos campos son opcionales. Además hay una serie de opciones que puedes marcar como pedir que te envíen los resultados por correo, que te manden notificaciones de eventos relacionados con spoj, etc.



**update user data**

User: programador01

Your name:

E-mail:

Show email to others: ☐

Email submission results to me: ☐

Notify me of important events and contests (max. 1 e-mail/month): ☒

Notify me of important news from SPOJ Team (e.g. recruitment proposals) (max. 2 e-mails/month): ☒

Remove me from the user ranking (main contest): ☐


Disable advanced submission editor: ☐

Disable source preview window on status page: ☐

New password:

Re-type new password:

Country:

Institution (optional):  

Motto/URL (optional):

Default programming language:

### 2.3.2. - Distribución interfaz

La página web tiene una distribución simple pero bastante amigable, consta de un menú lateral izquierdo fijo en el que se muestran todas las opciones de las que dispone la web, según la opción seleccionada cambia el contenido de la página.

Register: enlaza al formulario de registro del que ya he hablado.

Tutorials: esta página muestra enlaces a cuatro tutoriales sobre nuevos usuarios, problemas, problemas interactivos, problemas de concurso. También hay información de que es spoj y una pequeña guía de cómo empezar.

Status: muestra una tabla con los últimos envíos, en el punto 2.3.3 hablaré de ello con profundidad

Submit: enlace a la página de enviar un problema de la que ya he hablado.

Problems: muestra la lista de problemas, de la organización de ellos hablaré en el punto 2.3.4.

Search: es un cuadro de búsqueda de google.

news: es la página principal, muestra información de concursos, número de total de envíos, número de usuarios registrados, número de problemas publicados, total de problemas, número de instituciones afiliadas, número de concursos afiliados, número de lenguajes de programación.

Contest: este enlace te redirige a una página que consta de dos tablas una con los concursos que están actualmente activos, y la otra con concursos ya terminados, en ambos los campos son código del concurso, nombre, fecha de inicio y fecha de fin. Si pulsas en el nombre de un concurso que esté abierto te redirige a otra página similar a spoj pero con la información del concurso, sino simplemente te redirige a una página con el nombre del concurso pero sin información.

Ranks: Hay diferentes rankings, se muestran en forma de tabla según, usuarios, en este se valora los puntos obtenidos, además también se muestran la posición del ranking, las siglas del país, el nombre del usuario, cantidad de problemas resueltos y los puntos conseguidos, y el número de retos y los puntos conseguidos. Por países, similar al de los usuarios, etc.

Foro: tiene acceso a un foro, pero es independiente de la página

### 2.3.3. - Ver envíos en tiempo real

La página también dispone de una tabla que permite ver los envíos en tiempo real, dicha tabla consta del id del envío, la fecha del mismo, el nombre del usuario que lo ha enviado, el título del problema, el resultado, el tiempo de CPU, el tiempo en memoria y el lenguaje de programación.

Para que sea más intuitiva las filas de la tabla son de dos colores, verdes si el problema ha sido aceptado y naranja en cualquier otro caso.

La longitud de la tabla es de veinte filas, además hay cuatro páginas más por lo que podrías ver los últimos 100 problemas enviados.

### 2.3.4. - Clasificación de los problemas

La clasificación de los problemas se puede hacer de cinco formas, por el id del problema, por reto, por problemas para aficionados, problemas sencillos, lista de problemas de enigma.

Las tablas que muestran los problemas contienen las mismas columnas, id del problema, título del mismo, código (es el que hay que introducir para hacer el envío), número de usuarios que lo han resuelto y porcentaje de acierto.

## 2.4. - Olimpiada Informática Española (<http://olimpiada-informatica.org/>)

La Universidad Politécnica de Cataluña es la encargada, actualmente de organizarla. El idioma que se usa en la página por defecto es el castellano, aunque es posible modificarlo al catalán.

### 2.4.1. - Necesidad de registrar

Es necesario registrarse y autenticarse para hacer uso del total de las opciones disponibles, no obstante sin autenticarse podemos navegar por la página ya que la mayoría de las opciones las pueden ver tanto los usuarios registrados como los que no. Además, para facilitar las cosas, en el menú lateral izquierdo, en el cual aparecen las opciones, tienen un asterisco aquellas opciones en las que para acceder a ellas es necesario estar registrado.

Dichas opciones son:

- **Material:** se despliega un submenú para poder acceder a las secciones de manuales de programación, soluciones de problemas de pruebas anteriores (OIE 2008, 2009 y 2010, aunque no están todos los problemas), videotutoriales en C++, para ayudar a iniciarte en este lenguaje y un juego en el que los usuarios pueden crear programas para que compitan con otros jugadores.
- **Problemas:** En este enlace se puede acceder a la lista de los problemas, dispone de varias páginas, de esta forma se ven mejor los problemas disponibles.
- **Datos personales:** enlace que te permite no sólo ver tus datos, sino que también tienes de modificarlos, como por ejemplo la contraseña, ya que cuando te creas una cuenta te mandan ellos una aleatoria.

#### 2.4.1.1. – Formulario de registro

Para acceder al formulario de registro pulsaremos en el enlace que hay encima del login “*Date de alta en la web*”, éste nos mostrará un formulario con campos de texto para introducir los datos solicitados, los cuales son: nombre, primer apellido, segundo apellido, DNI, fecha de nacimiento, dirección, población, provincia, código postal, teléfono, email, el idioma (puedes seleccionarlo a través de un desplegable), también te piden información académica relativa a tu centro de estudio, decir si eres alumno o profesor (ambas opciones aparecen en un desplegable), si quieres o no participar en la OIE, curso, centro docente, dirección, población, provincia, código postal, teléfono, e-mail, por último, se muestran las bases del concurso y un desplegable para decidir si las aceptas o no.

No disponen de ningún mecanismo para saber si eres un bot o no, como captchas, preguntas de operaciones numéricas, etc.

### 2.4.2. - Distribución interfaz

La página consta de una columna lateral izquierda que se divide a su vez en cuatro apartados, el primero es el menú del sitio con las opciones

- Principal: que redirige a la página de bienvenida en la que se informa de los nuevos concursos, además de un enlace al foro y otro enlace al correo de la web y en el centro de la página se mencionan a los patrocinadores, así como información general de la Olimpiada Informática Española.
- Más información: muestra información sobre la Olimpiada Informática, así como información específica para alumnos y para los profesores.
- Concursos: se muestran información de los concursos activos así como de los ya finalizados, de estos últimos se muestra un ranking con los resultados de los tiempos y puntuación que obtuvieron los participantes.
- OIE 2012- OIE 2007: cada enlace muestra la normativa del concurso, así como información de cómo se organiza, fases del concurso, resultados de la fase on-line y de la fase final.
- El segundo apartado permite seleccionar el idioma (castellano o catalán), el tercero es el de login en el que se deben introducir el usuario y la contraseña, por último, en el cuarto apartado se encuentra la información de contacto, es decir el email de la persona a la que debes escribir en caso de error.

### 2.4.3. - Ver envíos en tiempo real

Esta página web, no permite ver los envíos que se están produciendo en tiempo real, no obstante sí puedes ver los envíos que has realizado. De hecho, cuando envías un problema, ves, a los pocos segundos el veredicto.


### 2.4.4. - Clasificación de los problemas

Los problemas se clasifican por 10 categorías y por concurso 32 concursos.

El enlace que se encuentra en el menú principal, cuya opción es “Problemas (con juez)” sólo muestra el título de problema, el cual es un enlace al enunciado.



**OIE - Olimpiada Informática Española**  
 Universidad Politécnica de Cataluña - Fundación Aula






**Opciones**

- » Principal
- » Más Información
- » Concursos
- » OIE 2012
- » OIE 2011
- » OIE 2010
- » OIE 2009
- » OIE 2008
- » OIE 2007
- » Preguntas Frecuentes
- » Fotos
- » Material
- » Ránking
- » Problemas (con Juez)
- » Datos personales
- » Enlaces de interés
- » Salida

**Problema ipv6**

**Enunciado**
**Envío**
**Resultados**

**Enunciado**

 PDF
  PS

**Juegos de prueba**

**Ejemplo de entrada 1**

```

1
5
ABCD:EF01:2345:6789:ABCD:EF01:2345:6789
2001:0DB8:0000:CD30:0000:0000:0000:0000
2002:0000:0000:00AA:0000:0000:0A00:0001
0000:0000:0000:0000:0000:0000:FFFF:FFFF
0000:0000:0000:0000:0000:0000:0000:0000
      
```

**Ejemplo de salida 1**

```

ABCD:EF01:2345:6789:ABCD:EF01:2345:6789
2001:DB8:0:CD30:0:0:0:0
2002:0:0:AA:0:0:A00:1
0:0:0:0:0:FFFF:FFFF
0:0:0:0:0:0:0:0
      
```



## 2.5 Jutge.org ([www.jutge.org](http://www.jutge.org))

Para concluir con el estudio del panorama en el ámbito que nos ocupa, rescatamos otro juez creado por Omer Giménez (creador también de OIE), en el que también han colaborado Salvador Roura y Jordi Petit. En este caso realizamos un repaso más ligero, ya que consideramos que este juez, no aporta detalles especialmente novedosos con respecto a los ya expuestos.

### 2.5.1 Distribución de la página

Lo primero que apreciamos en diferencia con el OIE, es el idioma, está en inglés y no se observa la opción de cambiarlo, como sucedía en la otra web.

La página dispone de una columna lateral izquierda, en ella está el cuadro para hacer login y los enlaces disponibles sin estar autenticado.

Tiene pocas opciones visible si no estás registrado, es más las únicas opciones disponibles son, enlace para acceder al formulario de registro, el cual te pide que rellenes ciertos campos como el nombre, el email, fecha de nacimiento, país, lenguaje(aquí sí que no da la opción de elegir entre varios idiomas, español, catalán, francés e inglés), un desplegable para elegir si estamos de acuerdo con la política de la web, y escribir el código de seguridad para comprobar que es una persona el que se está registrando.

Otro de los enlaces, es por si has olvidado tu contraseña, también hay otro enlace para usar una cuenta de prueba por si no te quieres crear una, para ello sólo pulsas el enlace y se rellenan automáticamente los campos de usuario y contraseñas

| Stat   | Problem                          | Title | View |
|--------|----------------------------------|-------|------|
| X60848 | 1-bit comparator                 |       |      |
| X81362 | 2-bit counter                    |       |      |
| X12031 | 2-input multiplexer              |       |      |
| X82780 | 2-input multiplexer (4-bit wide) |       |      |
| X64833 | 4-bit adder                      |       |      |
| X42916 | 4-bit adder/subtractor           |       |      |
| X35448 | 4-bit ALU                        |       |      |
| X61860 | 4-bit comparator                 |       |      |
| X35277 | 4-bit counter                    |       |      |
| X58456 | 4-bit incrementer                |       |      |
| X30847 | 4-bit multiplier                 |       |      |
| X42168 | 4-input majority voter           |       |      |
| X45339 | 4-input multiplexer              |       |      |
| P21459 | 50 x 50 != 250                   |       |      |
| X37276 | 7-segment digit                  |       |      |
| P56059 | A0001. Cool lines                |       |      |
| X74712 | Accumulator                      |       |      |
| X31534 | Add 7 bits                       |       |      |
| P88567 | Adding subrectangles             |       |      |
| P34279 | Add one second                   |       |      |
| P94143 | Aggressive rooks                 |       |      |
| P57928 | A grid game                      |       |      |
| P71916 | Anagrams                         |       |      |

El enlace para ver los exámenes, pese a que es visible sin estar logueado, te muestra un mensaje pidiendo que te identifiques.

Por último hablaré del enlace de los concursos, que sí es visible aunque no estés logueado, el enlace te muestra dos listas, uno con los concursos activos y otra con los que ya han

terminado, en ambas listas se muestra el nombre del concurso así como un botón para ver el ranking de dicho concurso, el ranking se muestra a modo de tabla y las columnas son, la posición del ranking, el nombre del usuario, tantas columnas como problemas haya habido, y para cada celda se muestra el veredicto, y la puntuación del participante.

Una vez autenticados podemos tener acceso a nuestro perfil, de esta manera podremos ver nuestros envíos, así como nuestro perfil y modificar los datos del mismo, como añadir una foto, por ejemplo.

### 2.5.2 Envíos en tiempo real

Cabe destacar que no hay una tabla, ni nada parecido para ver los envíos en tiempo real de los usuarios, lo más parecido es en el apartado de concursos, cuando accedes a un ranking además de la tabla con los participantes, también hay una lista en el que se muestran el nombre del usuario, el problema, el identificador del envío, el veredicto, el compilador y la fecha del envío.

## 2.6 FLOP (Free Laboratory Of Programming)

FLOP es un servicio web funcionando sobre Apache Tomcat, y el juez es un programa implementado en C. Ambos tienen licencia GPL.

Este sitio web es un laboratorio virtual para practicar la programación. Contiene una colección de problemas de programación y un corrector automático de programas (juez) para dichos problemas. Dos usos principales son posibles:

- Si eres un estudiante, aprendiz de programación, entra en el enlace "*Colecciones*" y encontrarás multitud de problemas de programación, que puedes resolver en C++, Java, Python, Pascal o Haskell. El sistema admite una solución y la revisa, indicándolo en el acto si es correcta o no...Este sistema es abierto y gratuito. No es necesario identificarse para poder usar el sistema.
- Si eres un profesor de programación, en el enlace "*Colecciones de usuario*" verás una lista de problemas para tus alumnos. El sistema prepara la colección que has elegido y la publica para tus alumnos. Te envía un e-mail con la URL en que está la colección, lista para tus alumnos, y también prepara un informe con la actividad desarrollada por los estudiantes.

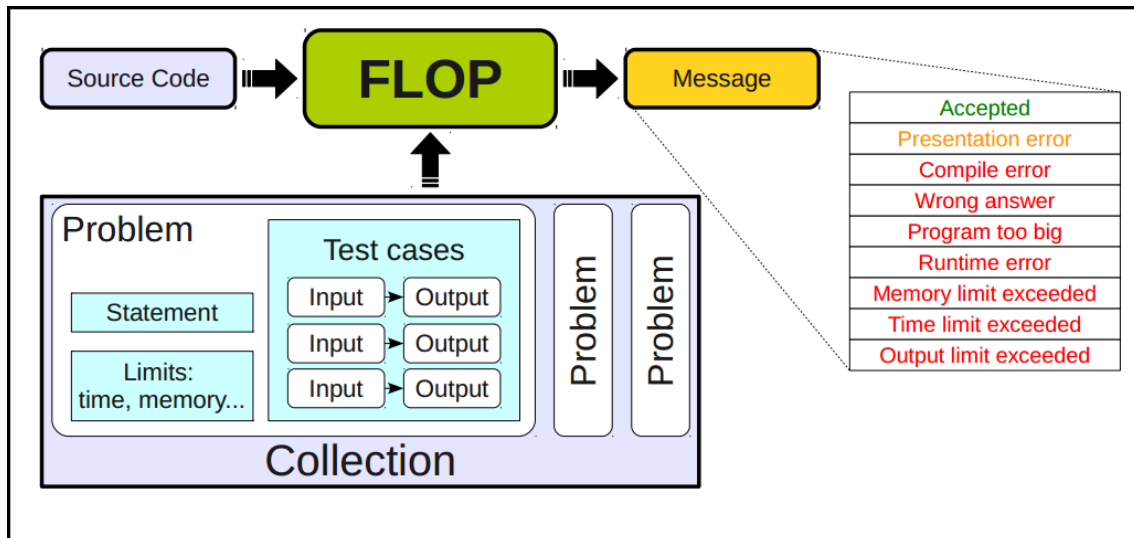
The screenshot shows the FLOP website interface. On the left is a sidebar with navigation links: Inicio, Área de prácticas, Área del profesor, Área del principiante, Lenguajes de programación soportados, Join & us, and Recursos. The main content area is titled 'FLOP — UN LABORATORIO LIBRE DE PROGRAMACIÓN' and 'COLECCIONES DEL PROFESOR'. It contains a paragraph explaining that a professor can prepare a problem sheet by selecting exercises from collections. Below this is a numbered list of three steps: 1. Give the collection name and email; 2. Manage the collection from the first link; 3. Give the second link to students. Further down, there are two sections: 'CREAR UNA COLECCIÓN A TU GUSTO' with input fields for 'e-mail' and 'Título de la colección' and a 'Crear una colección nueva' button; and 'RECUPERAR MIS COLECCIONES' with an 'e-mail' input field and a 'Recuperar mis colecciones' button. At the bottom left, there are logos for W3C XHTML 1.0 and W3C CSS.

Este sistema es gratuito y abierto para el profesor. No es necesario que te identifiques para poder usar este sistema.

El término "FLOP" es un simple acrónimo, que lo describe en inglés: "A Free Laboratory Of Programming". Ningún otro significado se le debe atribuir. Si quisiéramos anotar unas pocas palabras clave que lo describen, podríamos citar éstas: laboratorio de programación, abierto, gratuito, libre, ligero, de fácil uso y acceso... Desearíamos añadir también otras: útil, atractivo, disfrutar, divertido, aprender, etc.

Los lenguajes soportados actualmente son C, C++, Pascal, Java y Haskell.

La respuesta de FLOP es una de las siguientes: aceptado, respuesta incorrecta, error en ejecución, etc.



Los problemas están preparados minuciosamente siguiendo la plantilla típica de los concursos de programación más difundidos en el mundo (ACM ICPC, USACO, PKU, etc.); es decir: descripción general, descripción de la entrada y de la salida, un ejemplo de entrada y la salida correspondiente. Pero nuestros problemas no son de concurso, aptos sólo para alumnos brillantes, sino que abarcan los grados de dificultad típicos que se plantean durante el aprendizaje normal (desde triviales hasta muy difíciles). Los problemas están organizados en distintos bloques:

1. Introducción a la programación
  - a. Expresiones con datos básicos
  - b. Instrucciones de control: selección, iterativas
  - c. Subprogramas. Recursividad
  - d. Estructuras de datos: arrays, registros, etc.
2. Estructuras de datos y algoritmos (en desarrollo)

Y en cada bloque o sub-bloque se hallarán problemas de cinco niveles de dificultad, que podríamos calibrar informalmente así:

- Muy fáciles: sumar dos enteros, instrucciones de entrada/salida, convertir letras de mayúsculas a minúsculas, conversiones de valores de temperatura, cambio de monedas.
- Faciales: Obtener el siguiente número en la serie de Fibonacci, encontrar el término general de una matriz, cubrir el plano discreto, clasificar un ángulo.
- Medio: Encontrar la expresión de un número triangular, el máximo entre dos enteros, la condición para ser cuadrado perfecto.
- Difícil: Movimiento de un caballo del ajedrez, enumeración del domino, barajado perfecto.
- Muy Difícil: intervalos musicales, coordenadas de un triángulo, etc.

Esta catalogación es relativa al bloque. Por ejemplo, algunos de los ejercicios muy difíciles en el bloque de expresiones pueden quizá resolverse con mayor facilidad usando bucles.

## 3. - Arquitectura del Sistema

La arquitectura de nuestro juez evaluador se puede clasificar a nivel físico como la típica estructura cliente-servidor con tres niveles diferenciados, por un lado el acceso a datos, por otro el servidor con los servicios web, y por último la programación cliente. A continuación detallamos un poco más algunos aspectos de arquitectura.

### 3.1. Servicios Web

Un servicio web es una tecnología que se utiliza para intercambiar datos entre distintas aplicaciones mediante la utilización de un conjunto concreto de protocolos. El consumo de datos a través de un servicio web es independiente del lenguaje, lo cual resulta una ventaja para utilizarlos tanto desde PHP, como desde JavaScript (JS). En nuestro caso, los servicios web utilizados transmiten datos a través del protocolo REST, es decir de una interfaz web simple que utiliza XML y HTTP (y JSON).

La arquitectura REST, sigue unos principios fundamentales, que adoptamos en nuestro proyecto, tales como el uso de un conjunto de operaciones bien definidas (POST, GET, PUT, DELETE y OPTIONS), una sintaxis universal con la que poder direccionar cada recurso únicamente a través de su URI, y un protocolo cliente/servidor en el que cada mensaje HTTP contiene toda la información necesaria, (aunque se añade el uso de cookies para controlar detalles de la sesión).

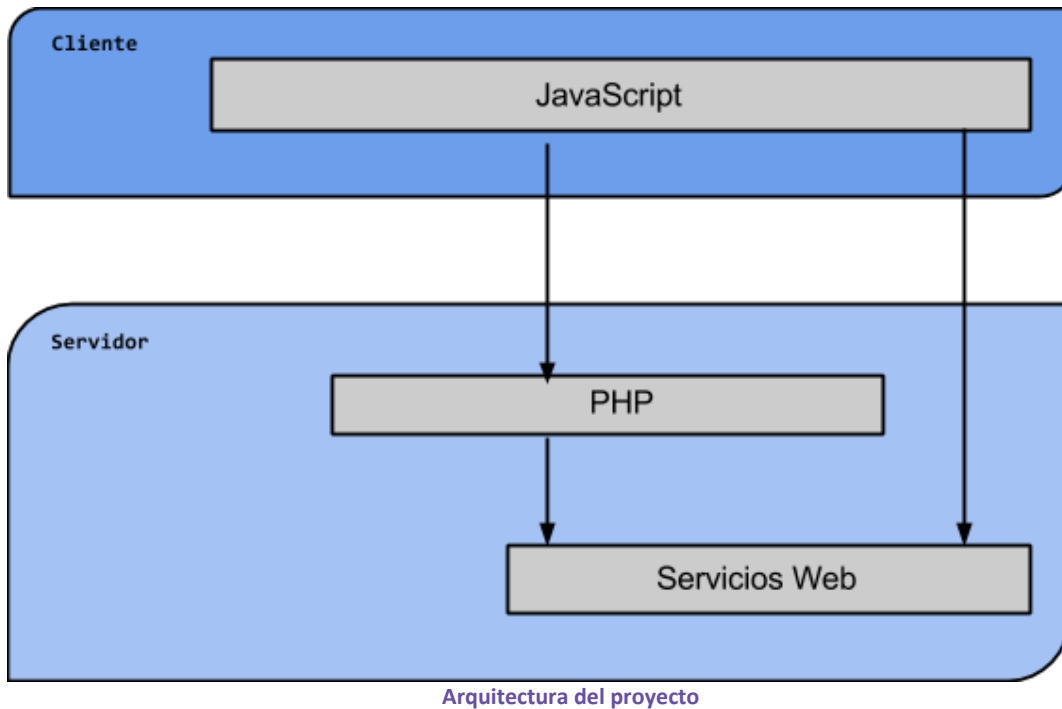
El consumo final de estos servicios se hará mediante las técnicas descritas en el apartado siguiente, y ampliadas en el apartado de Implementación.

### 3.2. PHP + JavaScript

Debemos distinguir programación del lado del servidor y programación del lado del cliente.

Como se ha comentado en otros apartados, en nuestro proyecto tenemos la necesidad de actualizar partes del contenido de la web de manera dinámica, sin necesidad de que el usuario actualice la ventana de su navegador. Para cumplir este requisito, es necesaria hacerlo del lado cliente. O en otras palabras, realizar las peticiones AJAX desde JavaScript. En nuestro caso, además, este tipo de peticiones asíncronas son utilizadas para solicitar datos a los servicios web que se mantienen en el servidor. En muchas ocasiones, estos datos los mostramos en forma de tablas, y las reconstruimos cada X segundos, estableciendo el periodo de refresco de antemano.

Este lenguaje es también especialmente útil para incorporar ciertos detalles en la web en “tiempo de navegación”, como efectos de animaciones, cambios dinámicos en el CSS o responder a eventos asociados a determinadas acciones del usuario como el click de un botón o la navegación entre campos de un formulario. Estos cambios, sólo tienen sentido si se hacen desde JavaScript.



Por otra parte, desde el lado servidor, se realiza otro tipo de programación. En este caso desde PHP, se implementan sobre todo aspectos de configuración de la web, tales como la gestión de la sesión de usuario, la gestión de las cookies, tratamiento de datos recogidos en formularios, o tratamiento de parámetros pasados a través de la URL.

Sin embargo, también realizamos peticiones a los servicios web desde el servidor con el conjunto de funciones curl (ver apartado de implementación). Estas peticiones se realizan antes de cargar la página web en el navegador, por lo que sólo deben hacerse en aquellos casos en los que la información que se necesita no va a cambiar, como por ejemplo el enunciado de un problema.

Aquí vemos un ejemplo de página que es construida por partes con diferentes tecnologías:

La interfaz de usuario de EDA muestra la siguiente estructura:

- Encabezado:** Logo de EDA y menú de navegación (Home, Problemas, Estadísticas, Documentación).
- Barra lateral izquierda:**
  - Autenticación: "Autenticado como: Don Quijote".
  - Mis últimos envíos: "logout" y "Ver tu perfil".
  - Formulario de búsqueda: "ID del problema" y botón "BUSCAR".
- Contenido principal:**
  - Perfil:** Sección con el título "Perfil".
  - DATOS USUARIO:** Tabla con los siguientes datos:
 

| DATOS USUARIO      |   |
|--------------------|---|
| Nombre:            | Don Quijote                                   |
| Nick:              | El de la triste figura                        |
| E-mail:            | a@b.com                                       |
| Género:            | Hombre  |
| País:              | España  |
| Rol:               | ADMIN   |
| Notificaciones:    | NEVER   |
| Fecha de registro: | registrado el 2012-11-26 a las 16:05:14+01:00 |
  - Problemas Aceptados (2/16):** Tabla con los siguientes datos:
 

| NÚMERO | TÍTULO                         |
|--------|--------------------------------|
| 20001  | El último dígito del factorial |
| 100    | The $3n + 1$ problem           |
- Pie de página:** Información de tiempo (10:39:58 GMT), derechos de autor (Public Domain / No Copyright) y créditos (Designed by...).

**Leyenda:**

1. Cabecera: La montamos directamente con HTML, ya que no cambia y no se construye con ningún WS.
2. Login: Se monta con PHP (en el servidor), comprobando la sesión actual.
3. Pie: Se monta con HTML imitando a la cabecera con la salvedad de que se usa JavaScript para mostrar el reloj.
4. Contenido: Se aprecian dos partes, bien diferenciadas
  - Los datos del usuario: que se piden y montan con PHP (lado servidor)
  - Los problemas aceptados, que se piden y montan con AJAX (lado cliente). Se refrescan con intervalos de tiempo diferentes según sea tu propio perfil o el de otro usuario, actualizando más rápidamente en el primer caso.

Así pues, debido al uso de estos dos tipos de tecnologías, en ocasiones se impone la comunicación entre ellas, para facilitar cierto tipo de implementaciones.

Todo esto, establece una arquitectura semejante a la que se ve en la figura, con la intercomunicación entre cliente-servidor y las distintas tecnologías.

### 3.3 TOMCAT

Lo primero que habría que destacar es que Tomcat no es un servidor de aplicaciones, sino un servidor web que tiene soporte de servlets y JSPs. Además, dispone del compilador Jasper, que es el encargado de compilar los ficheros JSPs y convertirlos en servlets. Es habitual que el motor de servlets de Tomcat se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. Al principio, se creía que el uso de Tomcat de manera autónoma, era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones, pero actualmente, Tomcat se usa como servidor web para entornos cuyo nivel de tráfico es alto.

Debido a que Tomcat ha sido escrito en Java, puede ser utilizado en cualquier sistema operativo, siempre y cuando disponga de una máquina virtual.

## 4.- Estudio de Navegación

A continuación detallamos las características del sitio web y sus opciones en cuanto a navegación se refiere.

Lo primero es entender qué queremos decir cuando nos referimos a la navegabilidad web, esto es, la facilidad con la que un usuario puede desplazarse por las distintas páginas web que componen el sitio. Para alcanzar un grado satisfactorio en la experiencia de navegación se deben adoptar ciertas estrategias diseñadas especialmente para tal fin, mejorando a la vez la usabilidad global de la aplicación.

Seguidamente analizamos y explicamos las decisiones tomadas a este respecto:

### 4.1. - Distribución general

Se ha realizado una división en partes del sitio con el objetivo de aunar el contenido bajo un mismo estilo que establezca la “seña de identidad” de la aplicación y consiga hacer sentir al usuario que entra en un espacio acotado.

Así pues, disponemos de una cabecera y un pie de página no cambiantes y un lateral adaptable al usuario y su estado en la web.

The screenshot displays the 'Programame!' website interface. At the top, a dark blue header contains the slogan '¡Acepta el reto!' on the left and a navigation menu with links 'Home', 'Problemas', 'Estadísticas', and 'Documentación' on the right. Below the header, the main content area is divided into two sections. The left section, with a light green background, contains a login form with fields for 'Usuario:' and 'Contraseña:', a 'Login' button, a link '¿Has olvidado tu contraseña?', and a 'Regístrate aquí' link. Below this is a search bar with the label 'ID del problema' and a 'BUSCAR' button. The right section has a white background and displays a welcome message: '!Bienvenido a Programame!' followed by 'Aquí va la presentación'. At the bottom of the page, a dark blue footer shows the time '9:24:41 (GMT)'.

En la cabecera, mantenemos un menú global con el que ofrecemos acceso desde cualquier página del sitio a la información más relevante de la aplicación. En concreto permitimos volver al índice, acceder a la lista de problemas (organizada por volúmenes), ver una tabla con los últimos envíos realizados (actualizada en tiempo real) y consultar documentación acerca del sitio ofreciendo algunas guías y consejos para ayudar a aquellos usuarios más desorientados.



Por otra parte, en el pie, además de los créditos sobre el estilo utilizado, incluimos un enlace “¿Quiénes somos?” que dirige a una página con algo de información sobre los “padres” de la aplicación, y un reloj que muestra la hora del servidor como referencia para visitantes foráneos.

Por último, pero no menos importante, incluimos un panel lateral en el que se recogen distintas funcionalidades:

- **Búsqueda:** Un usuario puede buscar un problema concreto desde cualquier página de la web, simplemente introduciendo su número de problema.
- **Login/información del usuario:** En el caso de un visitante no logueado, se mostrará un pequeño formulario para ingresar nombre de usuario y contraseña y autenticarse así, en el sitio, y un enlace al formulario de registro, para aquellos usuarios que aún no se hayan dado de alta en el sistema. Una vez logueado, sin embargo, este cuadro cambiará su contenido y pasará a mostrar un mensaje de autenticación, así como sendos links para ver sus últimos envíos, hacer logout, y ver su perfil. Para facilitar la navegación es importante hacer notar que la acción de loguearse te mantiene en la misma página para mayor comodidad.
- **Menús:** Dependiendo de la página que se visite, se mostrará un menú de navegación en la parte superior del lateral con enlaces a distintos contenidos relacionados con la página que se está viendo.

Con esta organización, dejamos un sector de la pantalla despejado para mostrar los contenidos en sí.

## 4.2. - Estadísticas (Últimos Envíos)

En esta página se muestra una tabla que se actualiza en tiempo real, con la información de los últimos 20 envíos de problemas que se han recibido. Esta tabla ofrece datos de la evaluación del problema, estado y demás, pero las columnas más relevantes en cuanto a navegación se refiere son las tres primeras:

- Num: Muestra el número del envío. Si el usuario está logueado y tiene además los permisos requeridos, se mostrará en forma de enlace a otra página que muestra el código fuente del problema resuelto en ese envío, facilitando así el acceso al mismo.
- User: Muestra el nick del usuario que ha hecho el envío. Dicho nombre será siempre un link al perfil del usuario, que mostrará su información.
- Problem: Muestra el título del problema resuelto en ese envío. De nuevo, este nombre será siempre un link a la página del problema.

Con estos enlaces, se mejora notablemente la usabilidad del sitio, además de resultar más práctico y cómodo, para labores de administración.

### ¡Acepta el reto!

- Últimos Envíos
- Rankings
- Globales

Usuario:

Contraseña:

Login

¿Has olvidado tu contraseña?  
 Regístrate aquí

ID del problema

BUSCAR

### Últimos Envíos

| NUM    | USUARIO        | TÍTULO PROBLEMA                    | RESULTADO | LENGUAJE | EJECUCIÓN | MEMORIA | CLASIFICACIÓN | FECHA DE ENVÍO            |
|--------|----------------|------------------------------------|-----------|----------|-----------|---------|---------------|---------------------------|
| 457539 | Ratulsaklain   | 10:6:2                             | IQ        | C        | ---       | ---     | ---           | 2013-06-12 a las 12:27:33 |
| 457538 | shantam.jain69 | Packing for Holiday                | AC        | C++      | 0.136     | 1272    | 150           | 2013-06-12 a las 12:27:28 |
| 457537 | Return Zero    | Fibonacci Numbers                  | AC        | C++      | 0.14      | 1272    | 46            | 2013-06-12 a las 12:27:15 |
| 457536 | GoneWind       | War                                | WA        | C++      | 0.14      | 1272    | ---           | 2013-06-12 a las 12:27:08 |
| 457535 | shantam.jain69 | One-Two-Three                      | AC        | C++      | 0.136     | 1272    | 99            | 2013-06-12 a las 12:26:56 |
| 457534 | wiragotama     | To Carry or not to Carry           | AC        | C++      | 0.136     | 1272    | 131           | 2013-06-12 a las 12:26:49 |
| 457533 | Gofranna       | Hashmat the Brave Warrior          | WA        | C++      | 0.144     | 1272    | ---           | 2013-06-12 a las 12:26:48 |
| 457532 | mukesh.meena   | Camel trading                      | WA        | JAVA     | 1.597     | 12736   | ---           | 2013-06-12 a las 12:26:08 |
| 457531 | Ratulsaklain   | 10:6:2                             | PE        | C        | 0.036     | 748     | ---           | 2013-06-12 a las 12:26:04 |
| 457530 | wiragotama     | To Carry or not to Carry           | WA        | C++      | 0.144     | 1272    | ---           | 2013-06-12 a las 12:26:01 |
| 457529 | Partha.CSE     | Peter's Smokes                     | WA        | C++      | 0.136     | 1272    | ---           | 2013-06-12 a las 12:25:59 |
| 457528 | kwunlunq       | Bicoloring                         | AC        | C++      | 0.132     | 1272    | 168           | 2013-06-12 a las 12:25:13 |
| 457527 | fagun.rain     | Leap Year or Not Leap Year and ... | WA        | C        | 0.036     | 748     | ---           | 2013-06-12 a las 12:25:02 |
| 457526 | wiragotama     | To Carry or not to Carry           | WA        | C++      | 0.14      | 1272    | ---           | 2013-06-12 a las 12:24:20 |
| 457525 | pjgblt         | TEX Quotes                         | WA        | C++      | 0.14      | 1272    | ---           | 2013-06-12 a las 12:24:16 |
| 457524 | prnb22         | Horror Dash                        | AC        | C        | 0.04      | 748     | 90            | 2013-06-12 a las 12:23:25 |
| 457523 | hoperecall     | Watering Grass                     | AC        | C++      | 0.14      | 1272    | 120           | 2013-06-12 a las 12:23:07 |
| 457522 | vjudge3        | Barcodes                           | WA        | C++      | 0.14      | 1272    | ---           | 2013-06-12 a las 12:22:49 |
| 457521 | PCCA_          | Internet Bandwidth                 | AC        | C++      | 0.14      | 1272    | 251           | 2013-06-12 a las 12:22:30 |
| 457520 | 6f70           | Playing with Wheels                | WA        | C++      | 0.14      | 1272    | ---           | 2013-06-12 a las 12:21:46 |

9:49:42 (GMT)

### 4.2.1. - Código enviado

Esta página sólo es accesible por administradores (para todos los envíos), o por los usuarios autores del envío que se está viendo. Esta funcionalidad es especialmente útil de cara a la adición de futuras funcionalidades relacionadas con la formación de clases, para que los profesores tengan fácil acceso al código de los alumnos.

¡Acepta el reto!

Home Problemas Estadísticas Documentación

Autenticado como:  
Don Quijote

Mis últimos envíos

logout Mi perfil

ID del problema

BUSCAR

Constante de Kaprekar --- ID 20002 --- Lenguaje: C++ --- Resultado: AC

```

#include <iostream>
#include <algorithm>
using namespace std;

void convert(int n, int &menor, int &mayor) {

    int v[4] = {0, 0, 0, 0};
    int pos = 0;

    // Pasamos al vector de dígitos.
    while (n != 0) {
        v[pos] = n%10;
        n = n / 10;
        pos++;
    }

    // Lo ordenamos
    std::sort(v[0], &v[4]);

    // Y construimos a y b
    menor = mayor = 0;
    for (int i = 0; i < 4; ++i) {
        menor *= 10; menor += v[i];
        mayor *= 10; mayor += v[3 - i];
    }
}

void resuelve() {

    int n;
    int cont = 0;

    cin >> n;

    while (n != 6174) {
        int a, b;
        convert(n, a, b);
        n = b - a;
        if (n == 0) {
            // repdigit!
            cout << 8 << "\n";
            return;
        }
        cont++;
    }

    cout << cont << "\n";
}

int main() {

    int cuantos;
    cin >> cuantos;

    for (int i = 0; i < cuantos; ++i)
        resuelve();

    return 0;
}

```

Comentario asociado a este envío:

Aquí puedes ver el comentario que asociaste a este envío, borrarlo, editarlo, o añadir uno nuevo si no lo has hecho ya.

500/500 caracteres

Editar

Borrar

9:53:16 (GMT)

## 4.3. - Problemas

En esta sección detallaremos la navegación por la web entre páginas con contenidos referentes a los problemas.

### 4.3.1. - Lista de Volúmenes

En esta página, mostramos una lista de los volúmenes que hay en el momento de la visita en forma de tabla. Los volúmenes se identifican por su número de volumen, que se mostrará dentro de un enlace, que dirige a la lista de problemas del volumen. Asimismo, se mostrarán unas estadísticas para que el usuario pueda hacerse una idea de lo fácil o difícil, que son los problemas incluidos en el volumen.

¡Acepta el reto!

Home Problemas Estadísticas Documentación

Autenticado como:  
Don Quijote

Mis últimos envíos

logout MI perfil

ID del problema

BUSCAR

Volúmenes

| ID  | AC/ENVÍOS   |
|-----|-------------|
| 1   | 13341/43389 |
| 2   | 4136/10972  |
| 3   | 5036/13959  |
| 4   | 11253/33620 |
| 5   | 6926/18983  |
| 6   | 5430/14237  |
| 7   | 2859/8704   |
| 8   | 2400/6988   |
| 9   | 1794/4310   |
| 10  | 900/3246    |
| 11  | 1169/3095   |
| 12  | 1346/3399   |
| 13  | 620/1321    |
| 14  | 416/1019    |
| 15  | 231/608     |
| 100 | 16361/49879 |
| 101 | 8613/28287  |
| 102 | 5532/16049  |
| 103 | 7626/19696  |
| 104 | 4361/11012  |
| 105 | 2383/6841   |
| 106 | 3784/8565   |
| 107 | 3344/8453   |
| 108 | 3120/9199   |
| 109 | 5285/13114  |
| 110 | 2875/7619   |
| 111 | 3834/8752   |
| 112 | 2419/6797   |
| 113 | 3571/9727   |
| 114 | 4720/13569  |
| 115 | 3799/9315   |
| 116 | 3111/7626   |
| 117 | 4233/9865   |
| 118 | 3816/7915   |
| 119 | 3896/10229  |
| 120 | 1086/2683   |
| 121 | 979/2093    |
| 122 | 1889/4133   |
| 123 | 1219/3029   |
| 124 | 1791/3702   |
| 125 | 3154/7489   |
| 126 | 1705/3949   |
| 200 | 14/110      |

¿Qué significan estos números? Los números A/T de cada volumen indican el número de envíos aceptados y el número de envíos totales entre todos los problemas de ese volumen. Son interesantes para hacerse una idea de la dificultad de un volumen en conjunto: cuanto más alto sea el porcentaje de envíos aceptados más fácil serán, normalmente, los problemas de ese volumen.

9:56:19 (GMT)

### 4.3.2. - Lista de Problemas

En esta página se muestra la lista de problemas, de nuevo en forma de tabla. Las columnas que se incluyen en esta ocasión son:

- ID del problema: Es un número identificador del problema. Cada volumen consta de 100 problemas, así que para ubicar el problema rápidamente se adopta una nomenclatura en la que a cada número de problema (0-99) se antepone el número de volumen (1-N). Así pues, el volumen 3 contiene los problemas 300-399, el volumen 7 los problemas 700-799, etc.
- Título: Es el título del problema. Tanto este campo, como el ID anterior son links al “mini-site” del problema. Se podría haber dejado un sólo enlace, pero de esta manera resulta más fácil llegar a la información del problema.
- Estadísticas de envíos: En esta columna se muestra una pequeña barra de progreso que muestra el estado del problema en referencia a la relación “*envíos con resultado AC (Accepted) / envíos totales*”.
- Estadísticas de usuarios: En esta columna se muestra una pequeña barra de progreso que muestra el estado del problema en referencia a la relación “*usuarios que tienen al menos un envío con estado AC (Accepted) / total de usuarios que ha intentado el problema*”.

¡Acepta el reto!

Home Problemas Estadísticas Documentación

Problemas

| ID    | TÍTULO                            | AC/ENVÍOS | AC/USUARIOS |
|-------|-----------------------------------|-----------|-------------|
| 20000 | Suma de dígitos                   | 0/0       | 0/0         |
| 20001 | El último dígito del factorial    | 4/36      | 2/2         |
| 20002 | Constante de Kaprekar             | 1/37      | 1/2         |
| 20003 | Encriptación de mensajes          | 0/1       | 0/1         |
| 20004 | Problemas de Herencia             | 1/9       | 1/1         |
| 20005 | Cuadrados diabólicos y esotéricos | 0/4       | 0/1         |
| 20006 | Móviles                           | 5/15      | 1/1         |
| 20007 | Ventas                            | 5/7       | 1/1         |
| 20008 | Códigos de barras                 | 0/1       | 0/1         |
| 20009 | Aproximación de Gauss             | 0/0       | 0/0         |

¿Qué significan estos números? La columna del centro muestra una pareja de números A/T indicando el número de envíos aceptados y el número de envíos totales de cada problema. Son interesantes para hacerse una idea de la dificultad del problema: cuanto más alto sea el porcentaje de envíos aceptados más fácil será, normalmente, un problema.

La columna derecha indica el número de usuarios que han conseguido resolver el problema, y cuántos lo han intentado. El porcentaje de aceptados en este caso indica la probabilidad de que, al final, resuelvas el problema; o, viéndolo de otro modo, el porcentaje de usuarios que, de momento, han dado al problema por imposible.

El número de aceptados en ambas columnas no tiene por qué ser igual. Si un usuario envía el mismo problema correctamente dos veces, contará por dos en la primera columna, y por uno en la segunda.

16:55:51 (GMT)

### 4.3.3. - “Mini-site” de un problema

Al seleccionar un problema, “descendemos un nivel” en la profundidad de la navegación, y la vista muestra un encabezado con el nombre del problema fijo para todas las páginas del mismo

### 4.3.3.1 - Enunciado

En esta página mostramos el enunciado del problema. Los enunciados contienen un formato concreto con las entradas y las salidas esperadas del problema. Por este motivo, lo formateamos con estilos distintos para resaltar el contenido y mejorar la comprensión de los enunciados.

## ¡Acepta el reto!

- Enunciado
- PDF
- Enviar
- Estadísticas
- Créditos del problema

Autenticado como: Don Quijote

Mis últimos envíos

logout Mi perfil

ID del problema

BUSCAR

### The Blocks Problem

Se trata de implementar un programa que sume los dígitos de un número entero no negativo. Por ejemplo, la suma de los dígitos del 3433 es 13.

Para darle un poco más de emoción, el programa no se limitará a escribir el resultado de la suma, sino que también escribirá todos los sumandos utilizados:  $3 + 4 + 3 + 3 = 13$ .

**Entrada**

La entrada consta de una serie de casos de prueba terminados con un número negativo. Cada caso de prueba es una simple línea con un número no negativo no mayor que  $10^8$  del que habrá que sumar todos sus dígitos.

**Salida**

Para cada caso de prueba se mostrará una línea en la que aparecerán cada uno de los dígitos separados por el signo +, tras lo cual aparecerá el símbolo = y la suma de todos los dígitos.

Ten en cuenta que antes y después de cada símbolo (+ y =) hay un espacio.

**Entrada de ejemplo**

```
3433
64323
8
-1
```

**Salida de ejemplo**

```
3 + 4 + 3 + 3 = 13
6 + 4 + 3 + 2 + 3 = 18
8 = 8
```

10:01:35 (GMT)

### 4.3.3.2 - PDF

En esta página, simplemente mostramos en una pestaña nueva el mismo enunciado en formato pdf, utilizando el visor del navegador.

The screenshot shows a PDF viewer interface. The main content area displays a document titled "Problemas de Herencia" with a URL "http://www.aceptaelreto.com/enunciado.php?num=0004". The document contains text in Spanish and three graphs labeled (a), (b), and (c). Graph (a) shows a function f(x) = x^2. Graph (b) shows a function f(x) = x^2 + 1. Graph (c) shows a function f(x) = x^2 + 1 with a shaded area under the curve. The text discusses the problem of finding the area under the curve f(x) = x^2 + 1 from x=0 to x=1.

#### 4.3.3.3 - Enviar

En esta página se realizará el envío del problema. Para ello se deberá elegir el lenguaje a usar en la implementación y posteriormente escoger entre introducir el código directamente en el editor incluido, o subir un fichero con la solución a entregar. El envío sólo será posible si el usuario está logueado. Además el envío de una solución a partir de un fichero, siempre tendrá prioridad, sobre una solución introducida en el editor, en caso de haber rellenado ambas cosas, y así se hará notar mediante un aviso al elegir subir un fichero.

#### 4.3.3.4 - Estadísticas

La página de estadísticas de un problema recoge información relevante del problema. En primera instancia se podrán ver un gráfico de tarta en tres dimensiones con los porcentajes de envíos hechos en los distintos lenguajes de programación; y un gráfico de barras (horizontal) con la cantidad de envíos de ese problema de las distintas evaluaciones posibles (AC, WA, CE, etc...). A continuación se muestra, en caso de haberse logueado, el mejor envío realizado para ese problema por el usuario que ha iniciado sesión. Por último, se dispone de una tabla con los veinte mejores envíos hechos para ese problema. La información que se muestra es: posición del envío, número de envío, usuario que efectuó mandó el problema con enlace al perfil del usuario, lenguaje usado en la implementación, tiempo de ejecución, memoria usada, y fecha de envío.

¡Acepta el reto!

[Home](#)
[Problemas](#)
[Estadísticas](#)
[Documentación](#)

Enunciado
PDF
Enviar
Estadísticas
Créditos del problema

Autenticado como: Don Quijote

Mis últimos envíos
logout
Mi perfil

ID del problema

BUSCAR

Constante de Kaprekar

Lenguajes

78.4%

C

C++

JAVA

Resultados

AC

WA

PE

TL

ML

OL

RF

RTE

CE

IR

0

10

20

30

40

Este es tu mejor envío de este problema

| Submission | Lang | Execution | Memory | Ranking | Date                      |
|------------|------|-----------|--------|---------|---------------------------|
| 437390     | C++  | 0.116     | 1272   | 1       | 2013-06-01T21:06:47+02:00 |

Clasificación

| POS | ENVÍO  | USUARIO                | LENGUAJE | EJECUCIÓN | MEMORIA | FECHA                     |
|-----|--------|------------------------|----------|-----------|---------|---------------------------|
| 1   | 437390 | El de la triste figura | C++      | 0.116     | 1272    | 2013-06-01T21:06:47+02:00 |

10:15:00 (GMT)


40



## 5. - Implementación

En esta sección se explicarán aquellos aspectos de implementación más relevantes, como librerías usadas, funciones importantes u otros detalles.

### 5.1. - jQuery

| jQuery                        |   |
|-------------------------------|---|
| jQuery 1.9 Released           |   |
| Desarrollador                 | <u>Equipo de desarrollo</u><br><u><a href="http://jquery.com/">http://jquery.com/</a></u> |
| Información general           |   |
| <u>Diseñador</u>              | <u>John Resig</u>   |
| Lanzamiento inicial           | <u>26 de agosto de 2006</u> ( <u>info</u> )   |
| <u>Última versión estable</u> | 1.9 ( <u>info</u> )<br>15 de enero de 2013; hace 2 meses                                  |
| <u>Género</u>                 | <u>Biblioteca</u>   |
| <u>Programado en</u>          | <u>JavaScript</u>   |
| <u>Sistema operativo</u>      | <u>Multiplataforma</u>  |
| <u>Licencia</u>               | <u>GPL</u> y <u>MIT</u>   |
| <u>Idiomas</u>                | Inglés (documentación)  |
| <u>En español</u>             |  *     |

**jQuery** es una biblioteca de JavaScript que facilita la forma de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y utilizar técnicas AJAX en páginas web.

**jQuery** es software libre y de código abierto, tiene un doble licenciamiento bajo la licencia MIT y la Licencia Pública General de GNU v2. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

### 5.1.2. Características (solo destacaremos las utilizadas en este proyecto)

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3. Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- Técnicas AJAX.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores funciones para rutinas comunes, etc.,

### 5.1.3. Uso

**JQuery** consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. Para mantenerlo en la última versión utilizamos un enlace donde JQuery mantiene actualizada la última versión.

```
<!-- SCRIPTS JS-->
<script src="http://code.jquery.com/jquery-latest.js"></script>
```

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Para ello utiliza las funciones `$()` o `jQuery()`.

### 5.1.4. Función `$()`

La forma de interactuar con la página es mediante la función `$()`, un alias de `jQuery()`, que recibe como parámetro una expresión CSS o el nombre de una etiqueta HTML y devuelve todos los nodos (elementos) que concuerden con la expresión. Esta expresión es denominada **selector** en la terminología de jQuery.

```
$("#tablaAlumnos"); // Devolverá el elemento con id="tablaAlumnos"
$(".activo");        // Devolverá una matriz de elementos con
class="activo"
```

```

switch (data.submission[0].result) {
    case "AC": $('#'+j+').append('<td class="aceptado">' + data.submission[0].result + '</td>');
                break;
    case "WA": $('#'+j+').append('<td class="error">' + data.submission[0].result + '</td>');
                break;
    case "IQ": $('#'+j+').append('<td class="encolado">' + data.submission[0].result + '</td>');
                break;
    case "CE": $('#'+j+').append('<td class="compilationError">' + data.submission[0].result + '</td>');
                break;
    case "IR": $('#'+j+').append('<td class="internalError">' + data.submission[0].result + '</td>');
                break;
    case "TL": $('#'+j+').append('<td class="timeLimit">' + data.submission[0].result + '</td>');
                break;
    case "ML": $('#'+j+').append('<td class="memoryLimit">' + data.submission[0].result + '</td>');
                break;
    case "RF": $('#'+j+').append('<td class="restrictedFunction">' + data.submission[0].result + '</td>');
                break;
    default: $('#'+j+').append('<td class="otroEstado">' + data.submission[0].result + '</td>');
                break;
}

```

Una vez obtenidos los nodos, se les puede aplicar cualquiera de las funciones que facilita la biblioteca.

```

// Se elimina el estilo (con removeClass()) y se aplica uno nuevo (con
addClass()) a todos los nodos con class="activo"
$(".activo").removeClass("activo").addClass("inactivo");

```

```

$('#gif').remove(); // Quitar la animacion del gif

```

| NUM | USER | PROBLEM | RESULT | LANGUAGE | EXECUTION | MEMORY | RANKING | DATE |
|-----|------|---------|--------|----------|-----------|--------|---------|------|
|-----|------|---------|--------|----------|-----------|--------|---------|------|



Con la función de la figura 2 quitamos el gif y en la figura 1 vemos un fragmento del código empleado para construir la tabla.

| NUM    | USER       | PROBLEM       | RESULT | LANGUAGE | EXECUTION | MEMORY | RANKING | DATE                      |
|--------|------------|---------------|--------|----------|-----------|--------|---------|---------------------------|
| 348428 | infomark12 | CD            | AC     | C        | 0.036     | 748    | 28      | 2013-05-08 a las 19:30:56 |
| 348427 | taharat    | Triangle Wave | AC     | C        | 0.036     | 748    | 124     | 2013-05-08 a las 19:30:50 |
| 348426 | vjudge10   | One-Two-Three | AC     | C++      | 0.14      | 1272   | 184     | 2013-05-08 a las 19:30:16 |
| 348425 | C^2        | Highways      | WA     | C++      | 0.136     | 1272   | ---     | 2013-05-08 a las 19:30:15 |
| 348424 | sepipe     | Jolly Jumpers | AC     | C++      | 0.14      | 1272   | 409     | 2013-05-08 a las 19:30:15 |

O por ejemplo, efectos gráficos:

```

// Anima todos los componentes con class="activo"
$(".activo").slideToggle("slow");

```

### 5.1.5. Función \$.getJSON ().

jQuery.getJSON(url[,][de datos, el éxito(datos, textStatus, jqXHR)]) devuelve jqXHR

Carga de datos JSON codificados desde el servidor mediante una petición HTTP GET.

jQuery.getJSON (url, [data] [, el éxito (datos, textStatus, jqXHR)])

#### Url

Tipo: Cadena- Una cadena que contiene la URL a la que se envía la solicitud.

#### Datos

Tipo: PlainObject- Un objeto simple o cadena que se envía al servidor con la solicitud.

#### Éxito (datos, textStatus, jqXHR)

Tipo: la función ()- Una función de devolución de llamada que se ejecuta si la solicitud se realiza correctamente.

Esta es una función Ajax equivalente:

```
$. Ajax ({
  dataType: "json" ,
  url: url,
  datos: datos,
  éxito: el éxito
});
```

```
$.ajax({
  async:false,
  cache:false,
  type:'POST',
  url: "ws/user/add/dry-run",
  data: '{"name":"' + document.form_registro.name.value + "','" +
    "nick":"' + document.form_registro.nick.value + "','" +
    "gender":"' + document.form_registro.gender.value + "','" +
    "email":"' + document.form_registro.email.value + "','" +
    "institutionId":"' + document.form_registro.institucion.value + "','" +
    "countryCode":"' + document.form_registro.country.value + "','" +
    "role":"' + "STUDENT" + "','" +
    "password":"' + document.form_registro.password.value + "','" +
    "mailNotification":"' + document.form_registro.mailNotification[marcado].value + "','" +
    "birthday":"' + fecha + "'" },
  contentType:'application/json; charset=utf-8',
  dataType:'json',
  success:function(response){
    respuesta = response;
    if (response.correct == "true") {
      document.getElementById("send").disabled = false;
    }
  }
});
```

| DATOS USUARIO      |   |
|--------------------|---|
| Nombre:            | Don Quijote                                   |
| Nick:              | El de la triste figura                        |
| E-mail:            | a@b.com                                       |
| Género:            | Hombre  |
| País:              | España  |
| Rol:               | ADMIN   |
| Notificaciones:    | NEVER   |
| Fecha de registro: | registrado el 2012-11-26 a las 16:05:14+01:00 |

Los datos que se envían al servidor se agregan a la URL como una cadena de consulta. Si el valor de los datos del parámetro es un objeto simple, se convierte en una cadena con codificación de URL antes de añadirlo a la URL.

La mayoría de las implementaciones utilizan un manejador de éxito:

```
$.getJSON( 'ajax / test.json' , la función (data) {
  var items = [];

  $. Cada uno (de datos, la función (clave, valor) {
    items.push( '<li id = "" + clave + ">" + val + '</ li>' );
  });

  $ ( '<ul/>' , {
    'Class' : 'my-new-list' ,
    html: items.join ( " " )
  }). AppendTo( "cuerpo" );
});
```

```
function tabla(){
  clearTimeout(timerID);
  $.getJSON(url_user + id + "/submissions", function(data){
    $('#gif').remove();
    var cell = document.getElementById("cont");
    if ( cell.childNodes() )
    {
      while ( cell.childNodes.length >= 1 )
      {
        cell.removeChild( cell.firstChild );
      }
    }
  });
}
```

## 5.2. AJAX

AJAX, acrónimo de Asynchronous JavaScript y XML, es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y DocumentObjectModel (DOM).

Tomaremos esta primera imagen como punto de partida para ver la evolución de una tabla que se actualiza con Ajax sin necesidad de recargar la página.

| NUM    | USER         | PROBLEM                    | RESULT | LANGUAGE | EXECUTION | MEMORY | RANKING | DATE                      |
|--------|--------------|----------------------------|--------|----------|-----------|--------|---------|---------------------------|
| 352482 | priya.nusrat | Hashmat the Brave Warrior  | IQ     | C        | ---       | ---    | ---     | 2013-05-10 a las 12:41:01 |
| 352481 | aaaaa        | Ugly Numbers               | IQ     | C        | ---       | ---    | ---     | 2013-05-10 a las 12:40:57 |
| 352480 | aaaaa        | Ugly Numbers               | IQ     | C        | ---       | ---    | ---     | 2013-05-10 a las 12:40:40 |
| 352479 | dhuacm2014   | Intersecting Line Segments | IQ     | C++      | ---       | ---    | ---     | 2013-05-10 a las 12:40:29 |
| 352478 | MaxistXXL    | Robots on Ice              | IQ     | C++      | ---       | ---    | ---     | 2013-05-10 a las 12:40:22 |

En esta segunda imagen hemos realizado un envío que marcamos en rojo.

| NUM    | USER                   | PROBLEM                   | RESULT | LANGUAGE | EXECUTION | MEMORY | RANKING | DATE                      |
|--------|------------------------|---------------------------|--------|----------|-----------|--------|---------|---------------------------|
| 352485 | El de la triste figura | Triangle Wave             | IQ     | C        | ---       | ---    | ---     | 2013-05-10 a las 12:42:59 |
| 352484 | wilson7356             | Triangle Wave             | IQ     | C++      | ---       | ---    | ---     | 2013-05-10 a las 12:42:32 |
| 352483 | dhiman10               | Love Calculator           | IQ     | C++      | ---       | ---    | ---     | 2013-05-10 a las 12:42:28 |
| 352482 | priya.nusrat           | Hashmat the Brave Warrior | IQ     | C        | ---       | ---    | ---     | 2013-05-10 a las 12:41:01 |
| 352481 | aaaaa                  | Ugly Numbers              | IQ     | C        | ---       | ---    | ---     | 2013-05-10 a las 12:40:57 |

Como podemos observar pasados 2 minutos nuestro envío pasa a ocupar la posición 5 de la tabla.

| NUM    | USER                   | PROBLEM           | RESULT | LANGUAGE | EXECUTION | MEMORY | RANKING | DATE                      |
|--------|------------------------|-------------------|--------|----------|-----------|--------|---------|---------------------------|
| 352489 | rajat08                | Above Average     | IQ     | C++      | ---       | ---    | ---     | 2013-05-10 a las 12:45:19 |
| 352488 | acmPrs                 | The Decoder       | IQ     | C        | ---       | ---    | ---     | 2013-05-10 a las 12:44:48 |
| 352487 | yanhanDebug            | Searching Quickly | IQ     | C++      | ---       | ---    | ---     | 2013-05-10 a las 12:44:21 |
| 352486 | wilson7356             | Triangle Wave     | IQ     | C++      | ---       | ---    | ---     | 2013-05-10 a las 12:43:45 |
| 352485 | El de la triste figura | Triangle Wave     | IQ     | C        | ---       | ---    | ---     | 2013-05-10 a las 12:42:59 |

Todas estas modificaciones se hacen sin recargar la página con una combinación de llamadas a funciones Ajax JQuery y JS.

### 5.2.1. Tecnologías incluidas en Ajax

Ajax es una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- DocumentObjectModel (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.

- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios. PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico también utilizado en el método Ajax.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

### 5.3. GeSHi – GenericSyntaxHighlighter(Marcador de sintaxis genérico)

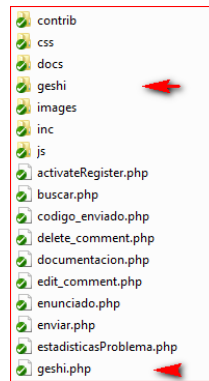
GeSHi. GeSHi comenzó como una idea para crear un marcador de sintaxis genérico para el sistema de foros phpBB, pero se ha convertido en un proyecto mucho más ambicioso. El objetivo de GeSHi es ser una herramienta de marcado simple pero a la vez poderosa, con las siguientes metas:

- Proporcionar soporte para un amplio rango de los más comunes lenguajes de programación.
- Facilitar el agregado de un nuevo lenguaje para el marcado
- Una amplia variedad de formatos para el usuario

GeSHi intenta llevar a cabo esta labor de la forma más ligera posible intentando que no sea un lastre en velocidad y rendimiento. La gran variedad de características de GeSHi facilitan un incremento de velocidad, de esta manera se puede encontrar un equilibrio entre la cantidad del marcado realizado y la velocidad con la que está hecho.

GeSHi da soporte a PHP5 y Windows, además de haber sido utilizado en el marcado de código en páginas ASP. Numerosos blogs, wikis y foros usan GeSHi, incluyendo Dokuwiki, Mambo, phpBB y WikkaWiki.

Para utilizarlo descargamos el código y lo metemos dentro del proyecto



Incluimos la clase php desde donde se hace uso de la funciones de la librería.

```
// Include the GeSHi library
include('geshi.php');
```

Utilizamos las funciones de la clase pasándole los parámetros adecuados.

```
$language = "JAVA";
if (isset($_GET['lang'])){
    if ($_GET['lang'] == "CPP")
        $_GET['lang'] = "C++";
    $language = $_GET['lang'];
}
$path = 'geshi/';

$geshi = new GeSHi($result, $language, $path);

// and simply dump the code!
echo $geshi->parse_code();
```

Este es el resultado final.

### Center of Masses --- ID 10002

```
// Autor: Nine2yL80594 UVA 943 Creo que esta vez dará Wrong answer
#include <iostream>
#include <stack>
using namespace std;

int main() {
    int n;
    while (cin >> n, n >= 0) {
        stack<int> p;
        do { p.push(n%10); n /= 10; } while (n > 0);
        int s = 0;
        while (!p.empty()) {
            s+=p.top();
            cout << p.top();
            p.pop();
            if (!p.empty()) cout << "+ "; else cout << " = ";
        }
        cout << 0 << endl;
    }
    return 0;
}
```

Código c++

### Puzzle --- ID 227

```
// Autor: Martin81172 UVA 163 Creo que esta vez dará Wrong answer
import java.util.*;

public class javawa {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);

        int n;
        do {
            n = in.nextInt();
            if (n >= 0) {
                int m = n;
                Stack<Integer> s = new Stack<Integer>();
                do {
                    s.push(m%10); m/=10;
                } while (m > 0);
                int a = 0;
                while (!s.empty()) {
                    a += s.peek();
                    System.out.print(a);
                    s.pop();
                    if (s.empty()) System.out.print(" =");
                    else System.out.print(" + ");
                }
                System.out.println(m + a);
            } while (n >= 0);
        }
    }
}
```

Código java



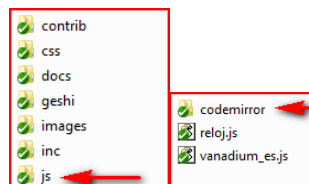
## 5.4. CodeMirror

CodeMirror es una librería externa que nos permite transformar cualquier elemento de nuestra página web en un editor de código integrado que soporta muchos de los lenguajes más utilizados en la actualidad.

Compatible con los navegadores Firefox, Chrome, Safari, Opera e Internet Explorer y con soporte de HTML, CSS, PHP, C, C++, C#, Java, Perl, PHP, JavaScript, Python, Lua, Go, Groovy o Ruby, con la ayuda de su API, el editor puede ser totalmente configurado.

CodeMirror incluye en su editor de código con licencia MIT el coloreado de sintaxis dependiendo del lenguaje en el que estemos escribiendo, autocompletado, vista previa de HTML,...

La forma de utilizarlo es bastante sencilla, primero necesitamos el código que lo podemos descargar de la página oficial, en nuestro caso lo hemos colocado dentro de la carpeta JS.



Una vez tenemos la librería dentro de nuestro proyecto enlazamos los distintos JS y CSS.

```
<!-- codeMirror (para quicksubmit) -->
<script src="/js/codemirror/lib/codemirror.js" type="text/javascript"></script>
<script src="/js/codemirror/mode/clike/clike.js" type="text/javascript"></script>
<script src="/js/codemirror/keymap/emacs.js" type="text/javascript"></script>
<link rel="stylesheet" type="text/css" href="/js/codemirror/lib/codemirror.css"/>
```

El ultimo paso es marcar un textarea como elemento para aplicar el codemirror y llamar a la funcion.

```
<p>Código:</p><textarea rows="15" cols="80" id="quick" name="quick" class="form-text" ></textarea>

<p>Archivo: </p>

<script>
  var editor = CodeMirror.fromTextArea(document.getElementById("quick"), {
    matchBrackets: true,
    lineNumbers: true,
    mode: "text/x-csrc",
    keyMap: "emacs"
  });
</script>

<script>
```

Este es el resultado final de la transformación del textarea en nuestro editor codemirror.

Lenguaje: C++

Código:

```

16      cout<<"La Hipotenusa es:"<<"\n"<<c<<endl<<endl;
17
18      cout<<"\t"<<"_\\ "<<endl;
19      cout<<"\t"<<"_\\ "<<endl;
20      cout<<"\t"<<"_\\ "<<endl;
21      cout<<"\t"<<"_\\ "<<endl;
22      cout<<"\t"<<"_\\ "<<endl;
23      cout<<"\t"<<"_\\ "<<endl;
24      cout<<"\t"<<a<<"_ "<<c<<endl;
25      cout<<"\t"<<"_\\ "<<endl;
26      cout<<"\t"<<"_\\ "<<endl;
27      cout<<"\t"<<"_\\ "<<endl;
28      cout<<"\t"<<"_ "<<b<<"_\\ "<<endl<<endl;
29
30
31      cin.ignore();
32      return 0;
33  }

```

## 5.5. VanadiumJS

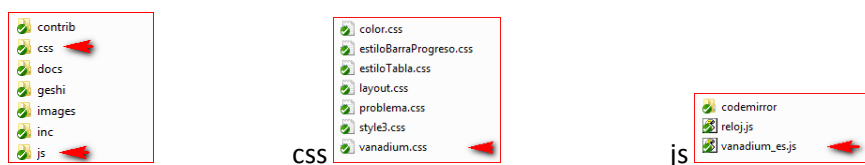
Valida el contenido de las entradas de texto (input text y check) de los formularios HTML. Por eso decimos que trabaja de la lado-cliente, pues verifica los datos según una condición específica y luego se envía los datos al servidor. Es una herramienta que trabaja con llamadas AJAX no-obstructivas haciendo uso de jQuery. Es una utilidad poderosa e intuitiva fácil de instalar y usar tal.

Respecto a la librería que hemos utilizado llamada VanadiumJS cabe destacar la sencillez de su configuración además de que es fácilmente personalizable para adaptarla a nuestras necesidades. Vamos a ver cómo funciona.

Para empezar, lo primero que haremos es explicar cómo hemos de configurar la librería e instalarla.

En la carpeta JS guardamos en primer lugar la librería de JQuery que podemos encontrar en su web es importante que también tengamos la librería de JQuery, en nuestro caso esta enlazada en la cabecera, como era de esperar hace uso de JQuery.

En la carpeta de CSS ponemos las hojas de estilo. Es una hoja de estilos sencilla de personalizar puesto que los nombres que llevan cada uno de los estilos son bastante intuitivos.



Finalmente en la cabecera enlazamos estos archivos.

```

<!-- <script type="text/javascript" src="/scripts/jquery-1.6.1.js"></script> -->
<script type="text/javascript" src="/js/vanadium_es.js"></script>

```

```

<!-- ESTILOS CSS -->
<link rel="stylesheet" type="text/css" href="/css/layout.css" />
<link rel="stylesheet" type="text/css" href="/css/color.css" />
<link rel="stylesheet" type="text/css" href="/css/estiloTabla.css"/>
<link rel="shortcut icon" href="/favicon.ico">
<link rel="shortcut icon" href="/favicon.ico">
<link rel="stylesheet" type="text/css" href="/css/style3.css" />
<link href="http://fonts.googleapis.com/css?family=Open+Sans+Condensed:700,300,300italic" rel="stylesheet" type="text/css">
<link rel="stylesheet" href="http://code.jquery.com/ui/1.9.2/themes/base/jquery-ui.css" />
<link rel="stylesheet" href="/resources/demos/style.css" />
<link rel="stylesheet" type="text/css" href="/css/vanadium.css" media="screen, tv, projection" />
<link rel="stylesheet" type="text/css" href="/css/estiloBarraProgreso.css" />
<link rel="stylesheet" type="text/css" href="/css/problema.css">

```

Vanadium usa el atributo *class* de los elementos de los formularios para que en él se escriban los ajustes que sean necesarios. Veamos un ejemplo con el que se validaría un campo de texto en el que se espera un e-mail.

```
<input id="email" name="email" type="text" class=":email"/>
```

Se puede observar que hay un símbolo de dos puntos (:) al principio de cada ajuste. En este caso estaríamos indicando que se realizara una validación según la expresión regular de un email. Si hacemos la prueba de este código veremos que realiza la validación por cada evento que ocurre en la caja de texto, si no queremos este comportamiento se puede cambiar para que lo realice pasado un tiempo después de un evento o para que lo realice sólo cuando el campo se abandone (terminar de escribir y pasar al siguiente campo). Otro aspecto importante es que en el ejemplo anterior se podría no escribir nada en la caja de texto y pasar la validación sin problemas, si se quiere que esto sea un campo requerido habría que usar otro ajuste.

```
<input id="email" name="email" type="text" class=":email :required : only_on_blur"/>
```

```

<td class="rtabform" ><label name="email">Email:</label>
</td>
<td id="celdaMail" class="segunda">
  <input name="email" id="email" type="text" class="form-text :email :required : only_on_blur" onblur="validarMail();" />
</td>

```

Email:  Por favor, introduzca una dirección de correo electrónico válida. Por ejemplo: xxx@domain.com .

Email:  Campo obligatorio.

Email:  Valor ya utilizado

Email:

También es posible la validación de otro tipo de campos a parte del email que va vienen implementados por defecto con la librería. Podemos encontrar en la página web del proyecto una lista de las posibilidades admitidas.

Veamos por ejemplo como validar un campo típico en el que se pide que repitas la contraseña, para comprobar que has escrito lo mismo en los dos lugares.

Contraseña

```
<input id="pass" name="pass" type="password" class=":password :required : only_on_blur"/>
```

Reescriba la contraseña por seguridad&lt;br /&gt;

```
<input name="repeatPass" type="password" class=":same_as;pass :required : only_on_blur"/>
```

```
<tr>
  <td class="rtabform"><label name="password">Contraseña:</label>
  </td>
  <td class="segunda" id="celdaPass">
    <input name="password" id="password" type="password" class="form-text :password :required : only_on_blur" onblur="validarPass();"/>
  </td>
</tr>
<tr>
  <td class="rtabform"><label name="pass2">Repite contraseña:</label>
  </td>
  <td class="segunda">
    <input name="pass2" id="pass2" type="password" class="form-text :same_as;password :required : only_on_blur" />
  </td>
</tr>
```

|                    |                          |                    |
|--------------------|--------------------------|--------------------|
| Contraseña:        | <input type="password"/> | Campo obligatorio. |
| Repite contraseña: | <input type="password"/> | Campo obligatorio. |

|                    |   |
|--------------------|---|
| Contraseña:        | <input type="password" value="*****"/>  |
| Repite contraseña: | <input type="password"/> Campo obligatorio.El valor debe ser el mismo que <u>Contraseña</u> . |

|                      |   |
|----------------------|---|
| Nombre:              | <input type="text" value="don quijote"/>  |
| Género:              | <input type="text" value="Hombre"/>   |
| Fecha de Nacimiento: | <input type="text" value="1933"/> <input type="text" value="Enero"/> <input type="text" value="1"/> |
| País:                | <input type="text" value="España"/>   |
| Institución:         | <input type="text" value="Universidad Complutense de Madrid"/>                                      |
| Email:               | <input type="text" value="a@c.com"/>  |
| Nick:                | <input type="text" value="ingenioso_hidalgo"/>  |
| Contraseña:          | <input type="password" value="****"/>   |
| Repite contraseña:   | <input type="password" value="****"/>   |

|                      |   |  |
|----------------------|---|--|
| Nombre:              | <input type="text"/>  | Campo obligatorio  |
| Género:              | <input type="text" value="Hombre"/>   |  |
| Fecha de Nacimiento: | <input type="text" value="2008"/> <input type="text" value="Enero"/> <input type="text" value="1"/> |  |
| País:                | <input type="text" value="España"/>   |  |
| Institución:         | <input type="text" value="Universidad Complutense de Madrid"/>                                      |  |
| Email:               | <input type="text" value="a@b.com"/>  | Valor ya utilizado   |
| Nick:                | <input type="text"/>  | Campo obligatorio  |
| Contraseña:          | <input type="password" value="*****"/>  |  |
| Repite contraseña:   | <input type="password"/>  | Campo obligatorio.El valor debe ser el mismo que <u>Contraseña</u> . |

## 5.6. Diagramas de google

Para una mejor interpretación de las estadísticas hemos incorporado dos tipos de graficas: de tartas y de barras. Para la implementación de estos diagramas hemos utilizado la una api de google llamada Google charts.

La api permite muchas opciones de configuración sobre las que no entraremos en detalles, basta con mencionarlalras.

Visualización: Gráfico de tartas

1. Visión de conjunto
2. Ejemplo
3. Carga
4. Formato de datos
5. Opciones de configuración
6. Métodos

7. Eventos
8. Política de datos

```

<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Task', 'Hours per Day'],
        ['Work',     11],
        ['Eat',      2],
        ['Commute',  2],
        ['Watch TV', 2],
        ['Sleep',    7]
      ]);

      var options = {
        title: 'My Daily Activities'
      };

      var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

```

<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Year', 'Sales', 'Expenses'],
        ['2004', 1000, 400],
        ['2005', 1170, 460],
        ['2006', 660, 1120],
        ['2007', 1030, 540]
      ]);

      var options = {
        title: 'Company Performance',
        vAxis: {title: 'Year', titleTextStyle: {color: 'red'}}
      };

      var chart = new google.visualization.BarChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

Estas imágenes muestran una forma simple de utilizar esta api para hacer los dos tipos de graficos de tartas y de barras.

Lo primero es enlazar la api a nuestra web.

```

<!--Load the AJAX API-->
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">

```

Una vez enlazado podemos utilizar las funciones de la api

```

// Load the Visualization API and the piechart package.
google.load('visualization', '1.0', {packages:['corechart']});

// Set a callback to run when the Google Visualization API is loaded.
google.setOnLoadCallback(drawChart);

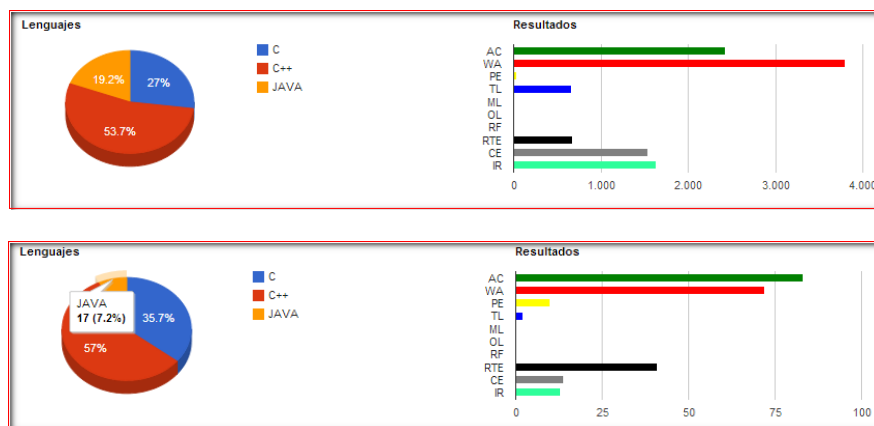
```

En la siguiente imagen vemos como se incrustan en el html los gráficos, previamente en data1 y data2 tenemos los datos que se quieren mostrar.

```
// Set chart options
var options2 = {
  title: 'Resultados',
  vAxis: {viewWindow: {min: 0}},
  colors: ['green','red', 'yellow','blue','orange','purple','#A9D0F5','black','grey','#2EFE9A'],
  legend: 'none',
  isStacked: true
};

// Instantiate and draw our chart, passing in some options.
var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
var chart2 = new google.visualization.BarChart(document.getElementById('chart_div2'));
chart.draw(data1, options1);
chart2.draw(data2, options2);
```

El resultado final es el siguiente:



## 5.7. Cross-site scripting

XSS, del inglés Cross-site scripting es un tipo de inseguridad informática o agujero de seguridad típico de las aplicaciones Web, que permite a una tercera parte inyectar en páginas web vistas por el usuario código JavaScript o en otro lenguaje script similar, evitando medidas de control como la Política del mismo origen. Es posible encontrar una vulnerabilidad XSS en aplicaciones que tenga entre sus funciones presentar la información en un navegador web u otro contenedor de páginas web. Sin embargo, no se limita a sitios web disponibles en Internet, ya que puede haber aplicaciones locales vulnerables a XSS, o incluso el navegador en sí.

XSS es un vector de ataque que puede ser utilizado para robar información delicada, secuestrar sesiones de usuario, y comprometer el navegador, subyugando la integridad del sistema. Las vulnerabilidades XSS han existido desde los primeros días de la Web.

Esta situación es usualmente causada al no validar correctamente los datos de entrada que son usados en cierta aplicación, o no sanear la salida adecuadamente para su presentación como página web.

Esta vulnerabilidad puede estar presente de las siguientes formas:

- Directa (también llamada Persistente): este tipo de XSS comúnmente filtrado, y consiste en embeber código HTML peligroso en sitios que lo permitan; incluyendo así etiquetas como `<script>` o `<iframe>`.
- Indirecta (también llamada Reflejada): este tipo de XSS consiste en modificar valores que la aplicación web utiliza para pasar variables entre dos páginas, sin usar sesiones y sucede cuando hay un mensaje o una ruta en la URL del navegador, en una cookie, o cualquier otra cabecera HTTP (en algunos navegadores y aplicaciones web, esto podría extenderse al DOM del navegador).

Para solucionar este problema hemos utilizado la función de php `htmlspecialchars()`

`htmlspecialchars` — Convierte caracteres especiales en entidades HTML

Descripción:

`string htmlspecialchars ( string $string [, int $flags = ENT_COMPAT | ENT_HTML401 [, string $encoding = 'UTF-8' [, bool $double_encode = true ]]] )`

Ciertos caracteres tienen un significado especial en HTML y deben ser representados por entidades HTML si se desea preservar su significado. Esta función devuelve un string con estas conversiones realizadas.

Si el string de entrada pasado a esta función y el documento final comparten el mismo conjunto de caracteres, esta función es suficiente para preparar entradas para su inclusión en la mayoría de los contextos de un documento HTML.

Las traducciones realizadas son:

- `'&'` (et) se convierte en `'&amp;'`
- `"""` (comillas dobles) se convierte en `'&quot;'` cuando `ENT_NOQUOTES` no está establecido.
- `"` (comilla simple) se convierte en `'&#039;'` (o `&apos;`) sólo cuando `ENT_QUOTES` está establecido.
- `'<'` (menor que) se convierte en `'&lt;'`
- `'>'` (mayor que) se convierte en `'&gt;'`

Repeating Decimals --- ID 202

esto es una prueba de xss

Comentario asociado a este envío:

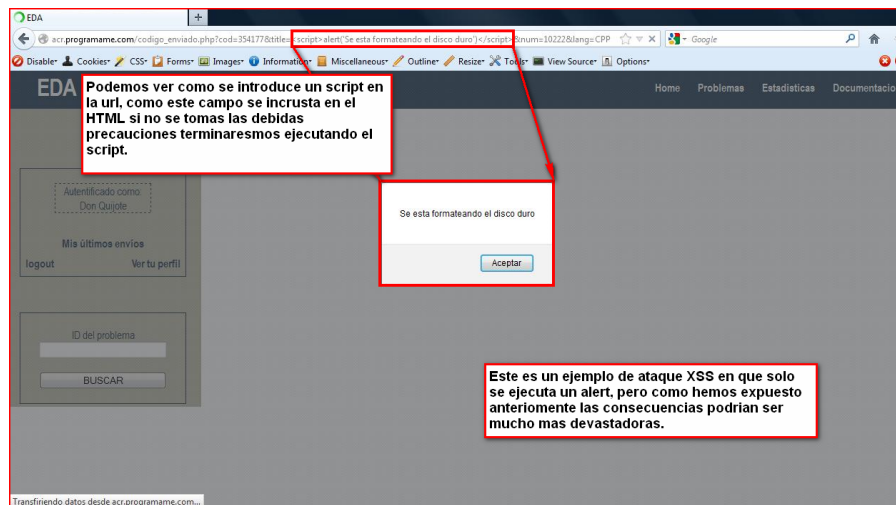
Aqui puedes ver el comentario que asociaste a este envío, borrarlo, editarlo, o añadir uno nuevo si no lo has hecho ya:

<script> alert('se esta formateando el disco duro')</script>

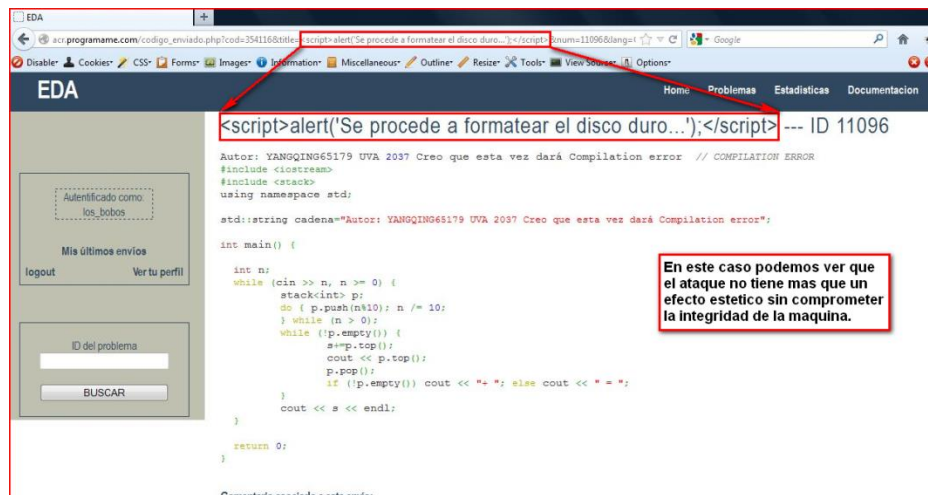
500/500 caracteres

Editar
Borrar

A continuación mostramos un ejemplo de un ataque de este tipo



En este otro caso hemos utilizado la función `htmlspecialchars`





## 5.8. cURL

Es una herramienta para usar en un intérprete de comandos para transferir archivos con sintaxis URL, soporta FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, FILE y LDAP. cURL soporta certificados HTTPS, HTTP POST, HTTP PUT, subidas FTP, Kerberos, subidas mediante formulario HTTP, proxies, cookies, autenticación mediante usuario+contraseña (Basic, Digest, NTLM y Negotiate para HTTP y kerberos4 para FTP), continuación de transferencia de archivos, tunneling de proxy http y muchas otras prestaciones. cURL es open source/software libre distribuido bajo la Licencia MIT.

Algunas de las funciones del curl que utilizamos en esta aplicación son las siguientes

- `curl_close` — Cierra una sesión cURL
- `curl_exec` — Establece una sesión cURL
- `curl_getinfo` — Obtiene información relativa a una transferencia específica
- `curl_init` — Inicia sesión cURL
- `curl_setopt` — Configura una opción para una transferencia cURL

```
curl_setopt($handle,CURLOPT_CUSTOMREQUEST,[PUT|DELETE|OPTIONS]);
```

Esta función la empleamos para invocar a los distintos servicios web que nos proporcionan información de la base de datos. La invocamos desde php para montar las páginas que requieren información de estos WS.

Ejemplos de uso:

```
$url=URL_user.$_SESSION["id"]."/submissions/problem/".$_GET["num"];
$ch = curl_init();
curl_setopt($ch,CURLOPT_URL,$url);
curl_setopt($ch,CURLOPT_RETURNTRANSFER,TRUE);
curl_setopt($ch,CURLOPT_CUSTOMREQUEST,"OPTIONS");
curl_setopt($ch,CURLOPT_HTTPHEADER,array("Accept:application/json"));
curl_setopt($ch,CURLOPT_COOKIE,'acrsession='.$_SESSION['token']);
$result = curl_exec($ch);
$jsonphp = json_decode($result,true,2048);
$httpCode = curl_getinfo($ch,CURLINFO_HTTP_CODE);
curl_close($ch);
```

En este ejemplo utilizamos el parametro OPTIONS

En este otro ejemplo solicitamos el HTML que contiene el enunciado de un problema

```
//CURL PARA CONSEGUIR EL CODIGO HTML

$ch = curl_init();
curl_setopt($ch,CURLOPT_URL,$URLenunciado);
curl_setopt($ch,CURLOPT_RETURNTRANSFER,TRUE);
$result = curl_exec($ch);
$httpCode = curl_getinfo($ch,CURLINFO_HTTP_CODE);
curl_close($ch);
```

### Repeating Decimals

Se trata de implementar un programa que sume los dígitos de un número entero no negativo. Por ejemplo, la suma de los dígitos del 3433 es 13.

Para darle un poco más de emoción, el programa no se limitará a escribir el resultado de la suma, sino que también escribirá todos los sumandos utilizados:  $3 + 4 + 3 + 3 = 13$ .

**Entrada**

La entrada consta de una serie de casos de prueba terminados con un número negativo. Cada caso de prueba es una simple línea con un número no negativo no mayor que  $10^8$  del que habrá que sumar todos sus dígitos.

**Salida**

Para cada caso de prueba se mostrará una línea en la que aparecerán cada uno de los dígitos separados por el signo +, tras lo cual aparecerá el símbolo = y la suma de todos los dígitos.

Ten en cuenta que antes y después de cada símbolo (+ y =) hay un espacio.

**Entrada de ejemplo**

3433  
64323  
8  
-1

**Salida de ejemplo**

3 + 4 + 3 + 3 = 13  
6 + 4 + 3 + 2 + 3 = 18  
8 = 8

## 5.9. MathJax

MathJax es una biblioteca javascript que permite visualizar fórmulas matemáticas en navegadores web, utilizando los lenguajes de marcado LaTeX o MathML. MathJax tiene licencia libre (licencia Apache, concretamente) y soporta múltiples navegadores.

MathJax, que es descargado entre el contenido de una página web, analiza el contenido de esta página para buscar expresiones matemáticas (en lenguaje LaTeX o MathML) y las dibuja. Por lo tanto, MathJax no requiere la instalación de ningún tipo de software o de tipos de letras adicionales en el sistema del lector. Esto permite que MathJax funcione en cualquier navegador con soporte JavaScript, incluidos los dispositivos móviles.

MathJax puede mostrar matemáticas utilizando una combinación de HTML y CSS o bien puede utilizar MathML, si es que está soportado por el navegador. El método exacto que MathJax utiliza para componer las expresiones matemáticas está determinado por las características del navegador del usuario, por los tipos de letras disponibles en el sistema del usuario y por la configuración que se haya establecido.

En caso de usar HTML y maquetación CSS, MathJax maximiza la calidad de las matemáticas mediante el uso de tipos de letras matemáticas, si están disponibles. En caso de que no lo estén, se recurre al uso de imágenes. En concreto, para los navegadores web más recientes, que pueden utilizar tipos de letras web, de hecho MathJax proporciona un amplio conjunto de tipos de letras web, que se descargan automáticamente, según sea necesario. Si el navegador no es compatible con tipos de letras web, MathJax comprueba si hay disponibles tipos de letras válidas en el sistema del usuario. Si esto no funciona, MathJax proporciona imágenes de los símbolos necesarios. MathJax se puede configurar para activar o desactivar el uso tipos de letras web, tipos de letras locales y de imagen.

MathJax puede usar notación matemática escrita en los lenguajes de marcado LaTeX o MathML. Debido a que MathJax está diseñado solamente para mostrar

matemáticas y LaTeX es un lenguaje de diseño de documentos, MathJax sólo es compatible con el subconjunto de LaTeX que se usa para describir la notación matemática.

Lo utilizamos para mostrar los enunciados de los problemas de forma más clara.

```
<script type="text/javascript" src="http://cdn.mathjax.org/mathjax/1.1-latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML-full"></script>
```

**Re-connecting Computer Sites**

Se trata de implementar un programa que sume los dígitos de un número entero no negativo. Por ejemplo, la suma de los dígitos del 3433 es 13.

Para darle un poco más de emoción, el programa no se limitará a escribir el resultado de la suma, sino que también escribirá todos los sumandos utilizados:  $3 + 4 + 3 + 3 = 13$ .

**Entrada**

La entrada consta de una serie de casos de prueba terminados con un número negativo. Cada caso de prueba es una simple línea con un número no negativo no mayor que  $10^9$ . Hay que habrá que sumar todos sus dígitos.

**Salida**

Para cada caso de prueba se mostrará una línea en la que aparecerán cada uno de los dígitos separados por el signo +, tras lo cual aparecerá el símbolo = y la suma de todos los dígitos.

Ten en cuenta que antes y después de cada símbolo (+ y =) hay un espacio.

**Entrada de ejemplo**

```
3433
64323
8
1
```

**Salida de ejemplo**

```
3 + 4 + 3 + 3 = 13
6 + 4 + 3 + 2 + 3 = 18
8 = 8
```

Para poder realizar el cálculo utilizaremos la solución que aportó el famoso matemático Riemann. Riemann asegura que se puede aproximar el área que se encuentra limitada superiormente por una función por las llamadas *sumas de Riemann*. El método consiste en considerar pequeños rectángulos todos del mismo ancho y cuya altura se corresponde con el valor de  $f(x)$  de manera que el rectángulo toque en algún punto a la función. En nuestro caso, consideraremos que la toca en el *vértice superior izquierdo* (figura 1.b). Una buena aproximación del área total que hay por debajo de la función es la suma de todos esos pequeños rectángulos. Cuantos más rectángulos utilizemos, mejor será la aproximación (y más estrechos serán esos rectángulos). Observa que si tenemos  $n$  rectángulos, su anchura (base) es  $base_i = 1/n$ . Teniendo esto en cuenta, la aproximación del área total de tierra será:

$$A = \sum area_i = \sum base_i * altura_i = \sum (1/n) * altura_i = \sum_{i=0}^{n-1} (1/n) * f(i * (1/n))$$

1. Tiene las mismas cifras que el número de casillas. Es decir, siguen la serie de números naturales de 1 a  $n^2$ .
2. La suma de sus esquinas debe ser la *constante mágica 2* ( $CM2$ ) que cumple que:

$$CM2 = \frac{4 * CM}{n}$$

3. Si  $n$  es impar:
  - o La suma de las cifras de las cuatro casillas de la mitad de los laterales suman la *constante mágica 2*.
  - o Si se multiplica el valor de la casilla central por 4, se obtiene la *constante mágica 2*.

|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| 22 | 47 | 16 | 41 | 10 | 35 | 4  |
| 5  | 23 | 48 | 17 | 42 | 11 | 29 |
| 30 | 6  | 24 | 49 | 18 | 36 | 12 |
| 13 | 31 | 7  | 25 | 43 | 19 | 37 |
| 38 | 14 | 32 | 1  | 26 | 44 | 20 |
| 21 | 39 | 8  | 33 | 2  | 27 | 45 |
| 46 | 15 | 40 | 9  | 34 | 3  | 28 |

$$n = 7$$

$$\text{Constante mágica} = 175$$

$$\text{Constante mágica 2} = 100$$

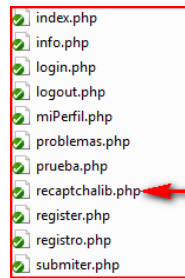
$$\text{Esquinas} \quad 22 + 4 + 46 + 28 = 100 \quad (CM2)$$

$$\text{Centro} \quad 4 * 25 = 100 \quad (CM2)$$

$$\text{Centro lados} \quad 41 + 13 + 37 + 9 = 100 \quad (CM2)$$

## 5.10. RECAPTCHA

La librería reCAPTCHA nos permite de forma simple incluir un CAPTCHA en nuestra página php, esto nos ayuda a prevenir ataques de bots. La librería la podemos descargar de la web oficial, es una clase php que tenemos que incluir en nuestro proyecto.



Para mostrar el captcha utilizamos un script que enlazamos directamente con la url que nos da el fabricante.

```
//cargamos el recaptcha
$.getScript("http://www.google.com/recaptcha/api/js/recaptcha_ajax.js",function(){
//creamos y mostramos el recaptcha
Recaptcha.create("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "recaptcha", {
    theme: "red",
    callback: Recaptcha.focus_response_field
  });
});
```

Una vez enlazado el script que mostrara el captcha tenemos que reservamos un espacio en el HTML para que se muestre, la forma de hacerlo es asignando el nombre a los ids y etiquetas correspondientes

```
<tr>
  <td class="rtabform"><label name="captcha">Captcha:</label>
  </td>
  <td class="segunda">
    <div id="recaptcha"></div>
  </td>
</tr>
```

El resultado es el siguiente:



Este es el aspecto que tiene dentro del formulario, si no es claro el texto podemos recargar el captcha para que nos muestres otros resultados



Con esto ya tenemos montado el Captcha pero tenemos que validar el resultado, para esto en la página a la que enviamos el formulario hacemos uso de la librería que antes nos descargamos e incluimos en el proyecto.

```
//RECAPTCHA
require_once('recaptchalib.php');
require_once('Urls.php');

extract($_POST);

$privatekey = "*****";
$resp = recaptcha_check_answer ($privatekey,
$_SERVER["REMOTE_ADDR"],
$_POST["recaptcha_challenge_field"],
$_POST["recaptcha_response_field"]);
```

Sobre la variable \$resp podemos utilizar los métodos de la librería para saber si la respuesta es correcta.

```
if (!$resp->is_valid) {

    header ("Location: registro.php");

}
else{
    ....
    ....

    // CONTINUAMOS CON EL ENVO DEL FORMULARIOS
    ....
}
```

## 6. – Conclusiones

### 6.1. – Opiniones y valoraciones

A continuación presentamos las opiniones de algunos profesores de la Facultad de informática (UCM) que han utilizado la aplicación (omitimos sus nombres).

**Credenciales (nombre, apellido, asignatura, cargo...)**

Profesor de Fundamentos de la programación (1º curso de los grados)

**Opinión del proyecto**

Un proyecto muy interesante desde el punto de vista académico. Aunque todavía le falta funcionalidad, se atisba lo que se quiere conseguir y puede llegar a ser una herramienta muy útil.

**Aplicación docente**

Claramente su aplicación docente es inmediata en asignaturas de programación, que en nuestra facultad abundan. Muchos estudiantes podrían verse beneficiados de disponer una herramienta de evaluación automática de sus programas. Y los profesores podrán usar la herramienta para proponer nuevos ejercicios a sus estudiantes.

**Ventajas**

Su facilidad de uso, así como la posibilidad de realizar envíos en diferentes lenguajes.

**Inconvenientes**

---

**Predisposición a la implantación en sus clases**

Estaría perfectamente dispuesto a integrar una herramienta así en mis clases.

**Credenciales (nombre, apellido, asignatura, cargo...)**

Profesor de distintas asignaturas de los grados:

-Inteligencia Artificial

-Ingeniería del Conocimiento

-Sistemas Inteligentes

**Opinión del proyecto**

Creo que es un proyecto interesante y muy útil para la docencia de la programación

**Aplicación docente**

Puede servir para la evaluación de ejercicios en cualquier asignatura en la que la programación sea una componente importante. Y más interesante aún para la autoevaluación de los alumnos mientras están aprendiendo

**Ventajas**

Aparte de lo mencionado en el apartado anterior, añade el aspecto competitivo que es un incentivo muy importante para la mejora continua

**Inconvenientes**

Se echa en falta algo de documentación y leyendas para los veredictos. No se trata de verdaderos inconvenientes sino de las limitaciones propias de una versión aun no completada

**Predisposición a la implantación en sus clases**

Podría utilizarlo para evaluar los algoritmos de búsqueda en el espacio de estados que se usan en Inteligencia Artificial

**Credenciales (nombre, apellido, asignatura, cargo...)**

Profesor de Estructura de Datos y Algoritmos, en 2º de los grados

**Opinión del proyecto**

Me parece muy útil para que los alumnos se animen a programar, y la utilidad aumenta en la medida en que la base de problemas utilizados sea lo suficientemente amplia. Aunque lo he utilizado poco, a primera vista lo único que “mejoraría” es que no sé a qué responden los volúmenes de problemas, por lo que a priori no sé dónde me interesa entrar para comenzar a resolver problemas.

**Aplicación docente**

Para una asignatura como EDA puede resultar una herramienta muy útil, pero para que los estudiantes no pierdan el tiempo debería ser posible dirigirlos hacia los volúmenes de problemas que interesen para un tema concreto.

Otra cuestión es que no se analiza la calidad del código, lo cual es importante en una asignatura como EDA.

**Ventajas**

Disponer de una amplia colección de problemas a resolver con un conjunto completo de casos de prueba que pongan a prueba la robustez (y la eficiencia) de su código.

**Inconvenientes**

Se pasan por alto todas las chapuzas que se suelen meter en el código para conseguir que funcione, aunque sea de manera eficiente. Y aunque no es un inconveniente realmente, se fomenta la idea de que el diseño de los algoritmos y la formalización no son especialmente importantes, lo cual es un poco contrario a lo que se intenta enseñar en una asignatura como EDA.

**Predisposición a la implantación en sus clases**

MUY alta.



**Credenciales (nombre, apellido, asignatura, cargo...)**

Profesor de asignatura Metodología de la Programación en los títulos de Ingeniería (la equivalente a Métodos Algorítmicos en la Resolución de Problemas en los grados) y de Programación II en el Grado en Estadística Aplicada.

**Opinión del proyecto**

El proyecto tiene bastante interés tanto desde el punto de vista docente como del trabajo de los alumnos que lo están desarrollando.

Aunque la versión del sistema que he probado todavía no está terminada, creo que es un sistema muy completo y que permite a los alumnos que lo han desarrollado aplicar técnicas muy diversas.

**Aplicación docente**

En las asignaturas de programación su utilidad es directa, porque permite a los alumnos aprender de forma autónoma. En relación con este aspecto, es importante que el alumno tenga una respuesta relevante del sistema para que pueda aprender de los errores cometidos. He añadido algunos comentarios respecto a esto en el apartado de Inconvenientes.

**Ventajas**

Aunque mi experiencia en el uso de jueces automáticos es limitada, creo que este sistema tiene características que lo hacen especialmente interesante. Por un lado, tener un servidor de corrección de programas basado en una cola de espera es interesante porque no bloquea el sistema si se quieren realizar múltiples envíos. En este caso, sería interesante tener más información del estado del servidor (en alguna de las pruebas que he realizado la única forma de saber que el servidor estaba parado ha sido esperar suficiente tiempo).

Por otra parte, es muy interesante disponer de información sobre el tiempo de ejecución y la memoria consumida por otras soluciones del mismo problema. En este caso, sería interesante poder filtrar la información de usuarios (por ejemplo, solamente del usuario con el que se ha realizado el envío, o un grupo reducido de usuarios). En las pruebas realizadas no era fácil ver el ejemplo enviado en el ranking debido al gran número de datos almacenados en el sistema.

**Inconvenientes**

El inconveniente principal para su uso como aplicación docente es que no dispone de un modo en el que se indique la salida del compilador en el caso de errores de compilación, o los casos en los que no funciona correctamente el programa en el caso de respuesta incorrecta. He mostrado recientemente un sistema similar a mis alumnos en un curso de programación. Aunque no he preguntado directamente a los alumnos su experiencia con el sistema, mi impresión es que su uso ha sido bastante reducido. La falta de información sobre los

problemas que ha encontrado el juez con los programas enviados creo que es determinante en su adopción como herramienta habitual por parte de los alumnos (y los profesores).

El uso de una cola de tareas pendientes es muy interesante, pero disminuye la interactividad con el usuario. Esto puede ser relevante en un entorno docente si hay muchas peticiones de corrección (por ejemplo el último día de entrega de prácticas).

Aunque supongo que está fuera del ámbito de este trabajo de fin de grado, sería muy interesante que tuviera un modo de uso para el profesor en el que se puedan añadir problemas de forma sencilla.

Por último, es muy importante disponer de información o documentación del sistema: los distintos estados que tiene un programa enviado, los posibles motivos por los que la respuesta puede ser incorrecta, etc.

Supongo que en la versión final del sistema esta documentación estará disponible.

#### **Predisposición a la implantación en sus clases**

Desde mi punto de vista, para su uso en cursos de programación es fundamental que el sistema tenga un modo en el que se pueda proporcionar información adicional al alumno sobre qué es lo que está fallando (qué caso no funciona, el mensaje del error de compilación, etc). Sin esta información, su utilidad se reduce a cursos de programación avanzados.

## **6.2. – Valoración personal**

El proyecto ha logrado cumplir todos los objetivos básicos que se habían propuesto, respetando los requisitos.

En los primeros momentos heredamos un prototipo implementado en su 80% en java script que desechamos tras un mes de evaluación. Esto supuso un gran reto ya que teníamos unos plazos muy ajustados de entrega debido a retrasos en el inicio de la asignatura. Volviendo la vista atrás y observando los resultados nos sentimos muy satisfechos con nuestro trabajo.

Hemos provisto al juez online de una interfaz agradable, intuitiva, amigable y sin exceso de florituras, utilizando las últimas tecnologías disponibles que dan a la aplicación una apariencia actual y dinámica. Nos hemos asegurado de dotar a la página de seguridad contra los ataques maliciosos conocidos.

Si analizamos las encuestas hechas a profesores de la facultad de informática de la UCM veremos que todos están muy dispuestos a utilizar la aplicación en sus clases. Esto corrobora lo expuesto en el primer párrafo de estas conclusiones, los requisitos se han cumplido.

Por otra parte si vemos la opinión de los estudiantes no todos están dispuestos a utilizar la aplicación, para algunos supondría un exceso de trabajo y una herramienta de control por parte de los profesores. En lo que todos coinciden es en que es una buena herramienta para autoevaluarse, sin tener que esperar a las correcciones del profesor.

Un punto en contra que ven los docentes es el hecho de no poder evaluar la calidad del código, desafortunadamente eso no se encontraba dentro de los requisitos iniciales del proyecto pero es una interesante línea de desarrollo futuro.

Aunque cuenta con algunos detractores dentro del alumnado podemos afirmar que “....” Será una herramienta tremendamente útil en la docencia, y prevemos un uso generalizado por parte de alumnos y profesores.

## 7. –Bibliografía

- **MOLLY E. HOLZSCHLAG. (2000). LA BIBLIA HTML 4. EDICIONES ANAYA MULTIMEDIA**
- **GERMÁN GALEANO GIL, JOSÉ CARLOS SÁNCHEZ ALONSO, PABLO DÍAZ MÁRQUEZ. (2009). HTML EDICIÓN REVISADA Y ACTUALIZADA 2009. EDICIONES ANAYA MULTIMEDIA**
- **TOMMY OLSSON & PAUL O'BRIEN. (2008). THE ULTIMATE CSS REFERENCE. SITEPOINT**
- **LUC VAN LANCKER. (2012). JQUERY EL FRAMEWORK JAVASCRIPT DE LA WEB 2.0. EDICIONES ENI.**
- **CLIFF WOOTTON. (2001). JAVASCRIPT PROGRAMMER'S REFERENCE. WROX PRESS**
- **DAVID FLANAGAN. (1998). JAVASCRIPT THE DEFINITIVE GUIDE. O'REILLY**
- **NICHOLAS C. ZAKAS, JEREMY McPEAK, JOE FAWCETT. (2006). PROFESSIONAL AJAX. WILEY PUBLISHING**
- **ESTEBAN TRIGOS GARCÍA. (2000). PHP 4. EDICIONES ANAYA MULTIMEDIA**
- **DAVID SKLAR. (2005). INTRODUCCIÓN A PHP 5. EDICIONES ANAYA MULTIMEDIA**
- **LUIS MIGUEL CABEZAS GRANADO. (2004). PHP 5 EL LENGUAJE PARA LOS PROFESIONALES DE LA WEB. EDICIONES ANAYA MULTIMEDIA.**